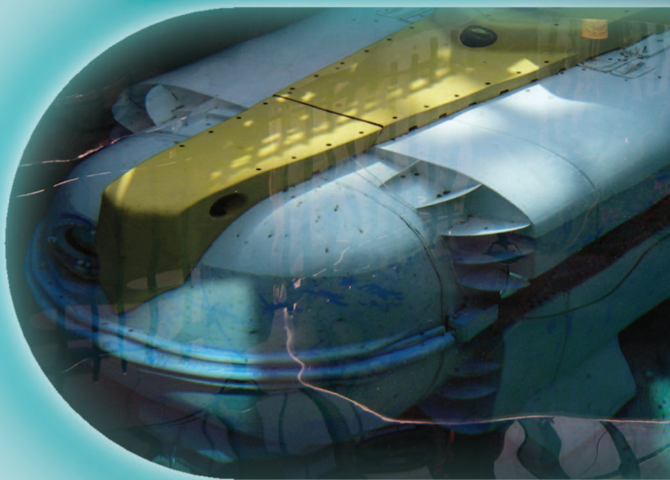
The Entrapment/Escorting Mission

An Experimental Study Using a Multirobot System

BY GIANLUCA ANTONELLI, FILIPPO ARRICHIELLO, AND STEFANO CHIAVERINI



Mobile Multirobot Systems

In recent years, multirobot systems have been the object of widespread research interest in the scientific community, given their application in different fields of robotics such as service, military, or educational robotics. The interest in using multirobot systems is due to their unique characteristics such as increasing the redundancy and the flexibility of mission execution, making the system tolerant to possible robot faults, accomplishing missions not physically executable by a single robot, or achieving the same mission of a single robot while reducing the execution time and increasing the performance. Moreover, the flexibility of multirobot systems is increased by the realization of systems with different typologies of autonomous vehicles such as wheeled mobile robots [15], [10], autonomous underwater vehicles [13], unmanned aerial vehicles [22], [6], and marine surface vessels [8], [16]. The research in multirobot systems has matured to the point where systems with hundreds of robots [18], [14] or teams of heterogeneous robots [9], [11] are being proposed.

Consistent research is devoted to applications such as exploration and mapping of unknown environments, pushing large objects, or studying biological systems, but few studies explicitly address the entrapment/escorting or catching problem. The entrapment/escorting mission consists in surrounding a moving target by reducing its escape windows (or, similarly, protecting a target by reducing the intrusion paths for an external agent) and can have different applications such as robotic surveillance security systems, military robotics, or entertainment robotics. In [17], a set of fuzzy rules are proposed to surround and entrap an escaping target, and these rules are experimentally validated on a three-robot system. In [23], an approach is presented to track and acquire a target and is experimentally validated by the use of two mobile robots.

From a control point of view, multirobot systems pose broadly different problems, such as motion planning and coordination, behavior emergence in unknown environments or unpredictable situations, information sharing, and the choice of sensor equipment. Among the possible control techniques, most control strategies for mobile robots resort to biologically inspired concepts, i.e., using elementary control rules of various animals (e.g., ants, bees, birds, and fishes) and trying to reproduce their group behavior (e.g., foraging, flocking, homing, and dispersing [19]) in cooperative robotic systems. Behavior-based approaches give the system the autonomy to operate in unpredicted situations using sensors to obtain information about the environment; thus, they are useful in guiding a multirobot system in an unknown or dynamically changing environment.

In this article, the entrapment/escorting mission is handled by resorting to the kinematic control presented in [4] and [5]. The proposed approach is based on a new kind of behavioral control, the null space-based behavioral (NSB) control [13]. This method differs from other existing behavioral coordination methods in the way that the outputs of the single elementary behaviors are merged to compose the final behavior. The NSB has been extensively tested in formation control missions [2], while in this article, its application to the entrapment/escorting mission is discussed. In particular, the control strategy has been validated both

Digital Object Identifier 10.1109/M-RA.2007.914932

 1070-9932/08/$25.00©2008 IEEE

in simulation and in several experimental case studies, where a team of six Khepera II mobile robots has to entrap a moving target represented by a tennis ball randomly pushed by hand. The simulative and experimental results show the effectiveness of the approach. Moreover, the control approach has been made robust such that in spite of the loss of a vehicle, in case of failure of one or more vehicles, the system autonomously reconfigures itself to correctly achieve the mission. Accordingly, in the experimental case studies shown, an intentional failure of one of the robots is imposed so as to show the structural robustness and the dynamic scalability property of the proposed technique with respect to the eventual loss of vehicles.

The NSB Control for Multirobot Systems

In a general robot mission, the accomplishment of several tasks at the same time is of interest. For instance, in a formation control mission, it is required that the vehicles maintain a given relative position while avoiding obstacles. A possible technique to handle the composition of the tasks has been proposed in [7], which consists in assigning a relative priority to single task functions by resorting to the task priority inverse kinematics introduced in [20] for ground-fixed redundant manipulators. Nevertheless, as discussed in [12], in the case of conflicting tasks, it is necessary to devise singularity robust algorithms that ensure proper functioning of the inverse velocity mapping.

Based on these works, this approach to the composition of the tasks has been developed in [4] in the framework of the singularity robust task priority inverse kinematics [12].

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

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

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.

An effective way of generating motion references $\boldsymbol{p}_{\mathrm{d}}(t)$ for the vehicles starting from the desired values $\boldsymbol{\sigma}_{\mathrm{d}}(t)$ 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., [24] for a tutorial). A typical requirement is to pursue a minimum-norm velocity, leading to a closed-loop inverse kinematics (CLIK) least-square solution:

$$\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{J}^{\dagger}(\dot{\boldsymbol{\sigma}}_{\mathrm{d}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1}(\dot{\boldsymbol{\sigma}}_{\mathrm{d}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}}), \tag{3}$$

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

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, based on the analogy of (3), the single task velocity is computed as

$$\boldsymbol{v}_i = \boldsymbol{J}_i^{\dagger}(\dot{\boldsymbol{\sigma}}_{i,\mathrm{d}} + \boldsymbol{\Lambda}_i\tilde{\boldsymbol{\sigma}}_i), \tag{4}$$

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, according to [12], the closed-loop solution (3) is modified into

$$\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1\right)\left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^{\dagger}\boldsymbol{J}_2\right)\boldsymbol{v}_3\right]. \tag{5}$$

The NSB control always fulfills the highest-priority task at nonsingular configurations. Remarkably, (5) has an agreeable 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 Escorting Mission

The mission of escorting a target can be seen as the requirement of surrounding a target whose movement is not known a priori but can be measured in real time. To achieve the mission, the multirobot system has to entrap the target and reduce its possible escape windows by properly distributing the team members around it. Thus, with reference to the planar case, the escorting mission can be satisfied by placing the n vehicles of the team at the vertices of a regular polygon of order n centered in the target and whose sides define a sort of intrusion/escape window (see Figure 1).

Following the NSB approach, the escorting mission is decomposed into elementary subproblems to be individually described and solved, which are as follows:

1) command the robots' centroid to be coincident with the target
2) move the robots on a given circumference around the centroid
3) properly distribute the robots along the circumference
4) avoid collisions among the robots themselves and with obstacles.

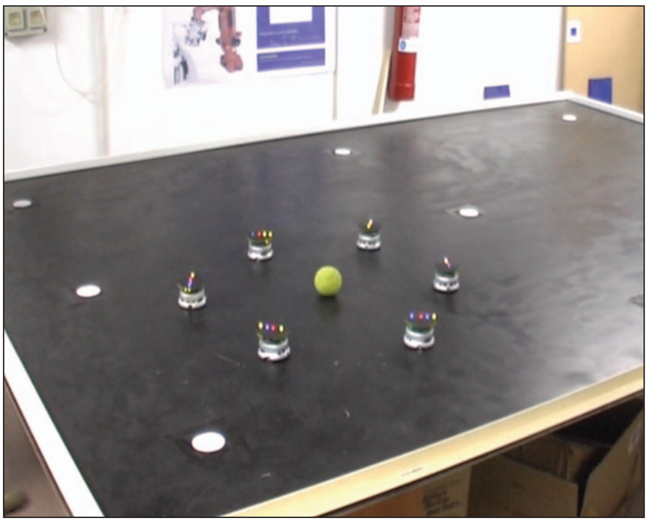


Figure 1. *The entrapment/escorting mission.*

Table 1. Selective activation, relative priority and CLIK gains of the behaviors in the five cases considered.

Task	Priority A	Priority B	Priority C	Priority D	Priority E	CLIK Gains
Centroid on the target	2	2	2	2	3	$\Lambda = 2.0$
Distribution on a circumference	–	3	–	3	2	$\Lambda = 0.5$
Polygon with equal edges	–	–	3	4	4	$\Lambda = 3.0$
Obstacle avoidance	1	1	1	1	1	$\Lambda = 1.0$

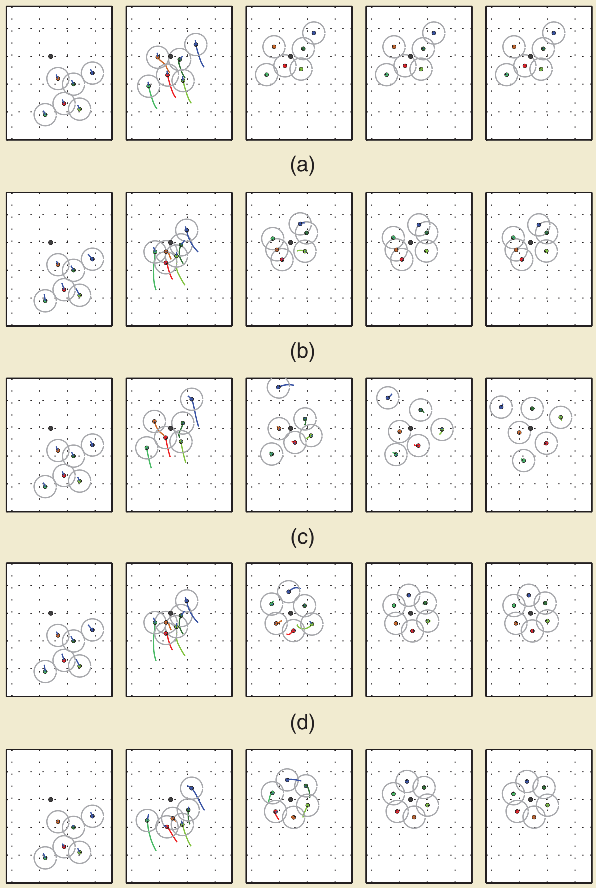


Figure 2. *Simulations of the entrapping mission with partial or full activation of the elementary behaviors: (a) obstacle + centroid; (b) obstacle + centroid + circular; (c) obstacle + centroid + polygon; (d) obstacle + centroid + circular + polygon; and (e) obstacle + circular + centroid + polygon.*

For each behavior, a suitable task function is properly designed. Without entering the mathematical details, which can be found in [5], the task function definitions are reported below.

1) For the centroid position, the two-dimensional task function $\boldsymbol{\sigma}_c$ is simply given by

$$\boldsymbol{\sigma}_c = \boldsymbol{f}_c(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p}_i = \bar{\boldsymbol{p}}. \tag{6}$$

2) The n-dimensional task function

$$\boldsymbol{\sigma}_s = \begin{bmatrix} \vdots \\ \frac{1}{2}(\boldsymbol{p}_i - \boldsymbol{c})^T(\boldsymbol{p}_i - \boldsymbol{c}) \\ \vdots \end{bmatrix} \tag{7}$$

can be used to keep each robot of the team at a given distance r from a point $\boldsymbol{c} \in \mathbb{R}^2$ by setting

$$\boldsymbol{\sigma}_{s,d} = \begin{bmatrix} \vdots \\ r^2/2 \\ \vdots \end{bmatrix}. \tag{8}$$

3) Properly distributing the robots along a given circumference is equivalent to making equal the relative distance between successive robots along this circumference. The latter task can be achieved by properly assigning the perimeter of the polygon inscribed in the circumference [5]. In fact, a regular polygon has the maximum perimeter among all the polygons of the same order inscribed on a given circumference. In this article, instead, the same configuration is pursued by requiring that the robots place themselves at the vertices of a polygon with sides of the same length. This is achieved by simply imposing the same distance between adjacent vehicles. It is worth noting that the task function definition used in this article has been shown to be more efficient than that proposed in [5] in the experimental runs.
4) The obstacle avoidance task function is defined individually for each vehicle, i.e., it is not an aggregate task function. In fact, each vehicle needs to avoid both environmental obstacles and the other vehicles. With reference to the generic vehicle of the team, in the presence of a punctual 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,

$$\sigma_o = \|\boldsymbol{p} - \boldsymbol{p}_o\|$$
$$\sigma_{o,d} = d,$$

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

According to (4), each elementary behavior outputs a velocity reference command to each robot of the team. To obtain the actual motion reference commands to the robots, the outputs that accomplish the single behaviors are merged by (5) on the basis of the active behaviors and on their priority orders.

Simulations

Extensive simulations have been performed with a selective activation of the behaviors and with different priority orders to better emphasize the meaning of each behavior and the importance of the priority orders. The simulations concern a team of nonholonomic robots that, starting from the same initial configuration, are commanded to accomplish different missions, depending on the active behaviors. It is worth noting that the simulation software uses the same control code realized to perform the experiments. Of course, in the simulations, instead of reading data from the camera vision system and sending data to the robots' actuators, the control code exchanges data with a kinematic simulator and a graphical interface. Besides simplifying the debugging of the control code, the simulator also allows the analysis of the behavior of the robots in ideal conditions that set the target performance to be pursued in the experiments. In particular, the absence of stochastic phenomena (e.g., measurement noise, variable delivery time, or loss of data in the radio communication) allows a repeatable comparison of different missions executed by starting from the same initial conditions.

The performed simulations concern five different situations denoted from A to E. Table 1 reports the active behaviors and their relative priority order for each case considered, and the CLIK gains are also given. For instance, in situation B, the highest-priority task is obstacle avoidance, the second-priority task is to keep the centroid of the team on the target, and the third-priority task is to distribute the robots on a circumference centered in the target. The task of placing the robots at the vertices of a regular polygon is not active. Remarkably, obstacle avoidance is always active and chosen as the primary task in all the missions to ensure safe execution of the mission.

Figure 2 reports several steps of the simulation for all the cases considered. In particular, Figure 2(a) shows the steps of mission A in which the robots have to keep their centroid on the target while avoiding collisions with it and among themselves. In this mission, the only control objective is the centroid. The shape of the robots thus remains uncontrolled. However, note that the final shape is not much different from the initial one. This can be explained by recalling that, among all the possible solutions for a single task, the NSB approach chooses at each step the one with the minimum velocity norm. As a consequence, the robots do achieve the mission, minimizing the motion in the null space of the centroid task function.

Figure 2(b) shows a mission (case B) in which the robots have to keep their centroid on the target and arrange themselves on a circumference of fixed radius. It is worth noting that the distribution along the circumference is uncontrolled, and thus the robots do not reach a regular polygonal shape. The addition of a behavior that places the robots at the vertices of a polygon with equal edges permits the accomplishment of the mission of entrapping the target. Figure 2(c) then shows the mission (case C) related to this elementary behavior, in which all the distances between adjacent vehicles surrounding the target are equal.

Cases D and E differ only in the order of priority of the active behaviors. The obtained simulation results are reported in Figure 2(d) and (e), which illustrate how the entrapment/escorting mission can be globally achieved by the use of the four proposed-task functions. Nevertheless, leaving out the obstacle avoidance behavior—the chosen elementary behaviors are not conflicting—all the tasks can be simultaneously solved at the end. Figure 2(d) and (e) shows that, at the last step, the target is surrounded by the vehicles that regularly distribute themselves around it. However, the different order of the priority of the tasks in the two cases changes the transient of the respective simulations.

Experiments

In the following section, the experimental setup and the results of the execution of several escorting missions with intentionally caused faults are reported.

Experimental Setup

The multirobot setup available at Laboratorio di Automazione Industriale of the Università degli Studi di Cassino, Italy, is composed of several Khepera II mobile robots manufactured by K-Team [1]. These are differentially driven mobile robots (with unicycle-like kinematics) with an approximate diameter of 8 cm. Each can communicate through a Bluetooth module with a remote Linux-based PC where the NSB has been implemented. To allow the needed absolute position measurements, we have developed a vision-based system using two CCD cameras, two

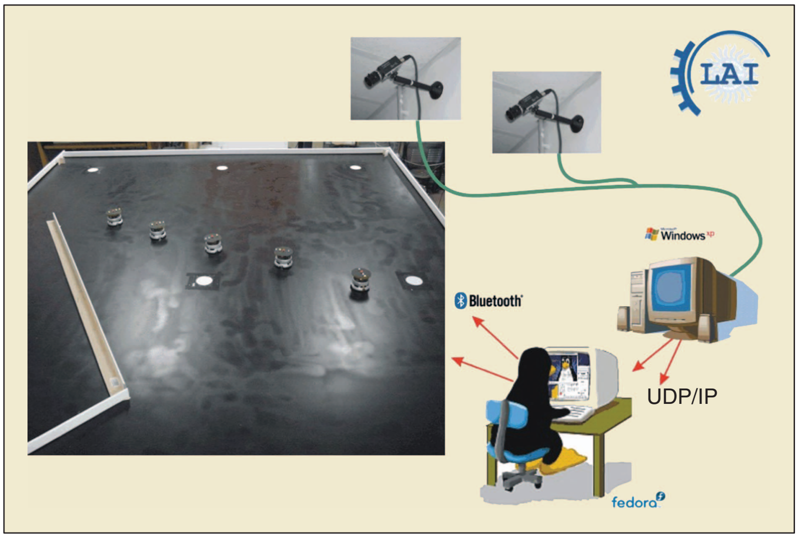


Figure 3. *Sketch of the multirobot setup available at Laboratorio di Automazione Industriale of the Università degli Studi di Cassino, Italy.*

Table 2. Order of priority and CLIK gains for the behaviors in the two experiments.

Task	Gain	Priority: First Experiment	Priority: Second Experiment
Obstacle avoidance	$\Lambda = 1.0$	1	1
Distribution on a circumference	$\Lambda = 0.5$	2	3
Centroid on the target	$\Lambda = 2.0$	3	2
Polygon with equal edges	$\Lambda = 3.0$	4	4

Matrox Meteor-II frame grabbers, and self-developed C++ image-processing functions. The acquired images are 1,024 × 768 RGB bitmaps. The measurement error has an upper bound of ≈0.5 cm and ≈1°. The remote PC, which implements the NSB control, receives the position measurements from the vision system at a sampling time of 100 ms. The NSB outputs the desired linear velocities for each robot, and, therefore, a heading controller is needed to obtain the wheels' desired velocities. We have developed a heading controller derived from the one reported in [21]. The remote PC sends (through the Bluetooth module) the wheels' desired velocities with a sampling time of $T = 80$ ms to each vehicle. The wheels' controller (onboard each robot) is a proportional integral derivative control loop developed by the manufacturer. A saturation of 40 cm/s and 100°/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 ≈9°/s is experienced. A sketch of the setup is shown in Figure 3.

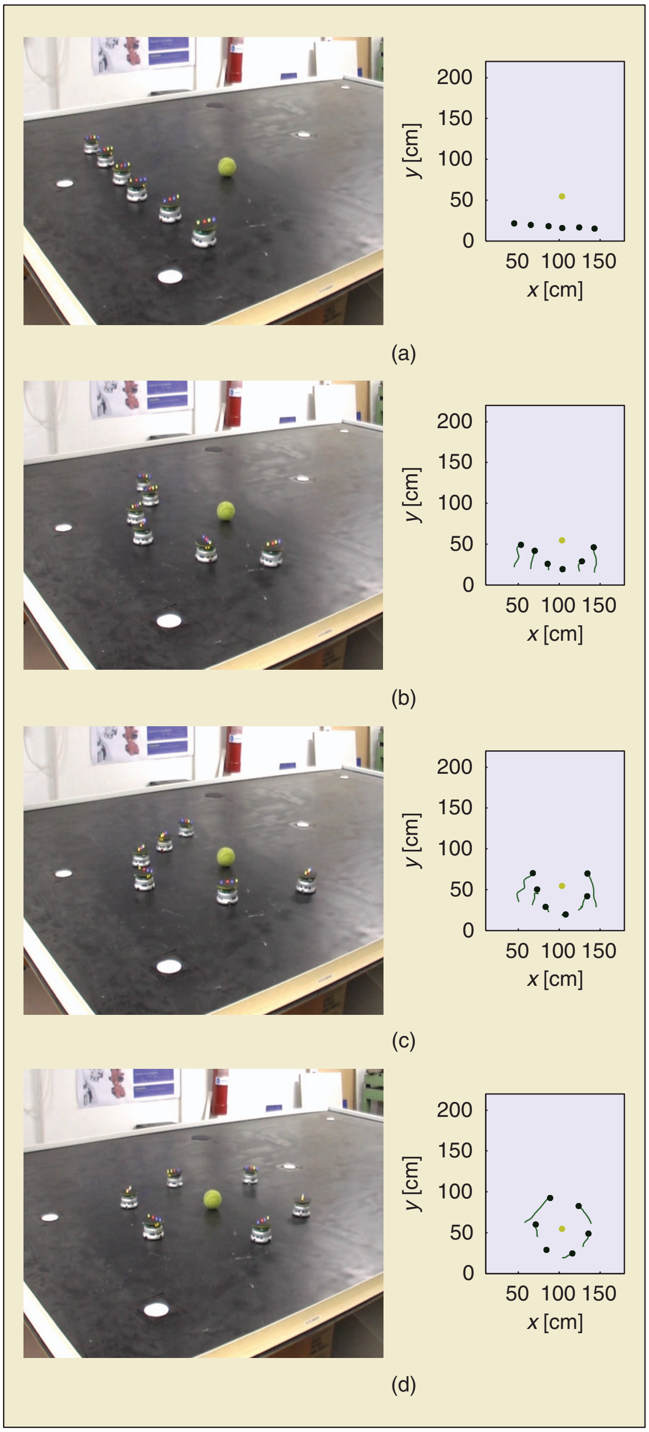


Figure 4. *First set of snapshots of the first escorting experiment: from $t = 0$ s to $t \approx 5$ s.*

Experimental Results

As a challenging case study, we report the experimental results of two different executions of a mission where a tennis ball is the target to be entrapped by a team of six Khepera II mobile robots. In particular, the vehicles should guarantee an escaping window of 40 cm while the safety distance imposed on the vehicles is 20 cm. The desired radius of the surrounding circumference is modified to guarantee the desired escaping window according to the number of robots, i.e., it is modified during the experiments to take into account the loss of one or more vehicles. To underline the effects of the task priority, the two executions differ only in the priority orders, while the topology of the mission and the task gains are exactly the same. The video images of the experiment are presented in two synchronized frames: the one on the left shows the videos acquired by a hand video camera, and the one on the right reports animations obtained using experimental data [25]. These animations are achieved through a self-developed C-based program that uses the OpenGL graphics library under the Linux environment.

For the first mission (the relative video is named `RAM_CIRCULAR.mpg`), we report a 30-s long section of the escorting mission. Initially, the ball is still, and the six robots have to surround it. Then, at $t \approx 6$ s one robot is moved away from the arena to simulate a failure, then it is put back in the arena at $t \approx 9$ s. Moreover, at $t \approx 16$ s, the target is pushed to impose a reconfiguration to the robots. The order of priority of the four tasks and the corresponding CLIK gains are summarized in Table 2. The video shows that the robots' positions in the circumference are not fixed a priori. After the failure of the robot, in fact, it is put back in the arena in a random position, and the platoon automatically reconfigures to include the recently added robot.

In Figure 4, the first five seconds are reported. The target is still, and the vehicles are required to surround it. It can be observed that the obstacle avoidance task is always the primary task, and the vehicles avoid hitting each other during the movement. Moreover, no predefined position is assigned around the target. A hexagon-like configuration is the natural structure of the six-robot formation since the regular polygon guarantees the minimum distance between adjacent points on a given circumference.

A fault is caused at $t \approx 6$ s by moving away a robot by hand and further obscuring it to the camera. The algorithm recognizes the absence of a robot as a major fault, i.e., the vehicle is lost, and the remaining robots have to complete the mission, ignoring the damaged robot and considering it as an obstacle. After the reconfiguration is successfully achieved, the robot is put again in the arena at $t \approx 9$ s. In Figure 5, the second group of snapshots are given, from $t \approx 5$ s to $t \approx 14$ s. Figure 5(a) and (b) shows the moment in which one of the robot is moved away. In Figure 5(b) and (c), the remaining vehicles are no longer minimizing the escape space of the target and need to reconfigure to achieve the lowest-priority task. From the geometrical point of view, it can be observed that this is achieved by positioning the vehicles from the vertices of a six-side regular polygon to those of a five-side regular one [Figure 5(b)] and modifying the desired radius accordingly. Moreover, when the vehicle is put back in the arena [Figure 5(c)] the formation is again rearranged into a hexagon. Note that, since the position of the robots in the formation is not specified, after recovering from the fault, the vehicle takes a different position from the one it had before the fault [Figure 5(d)].

The target is pushed twice to demonstrate that the algorithm is working in real time, and the vehicles reconfigure such that the escort mission is still accomplished. This can be seen

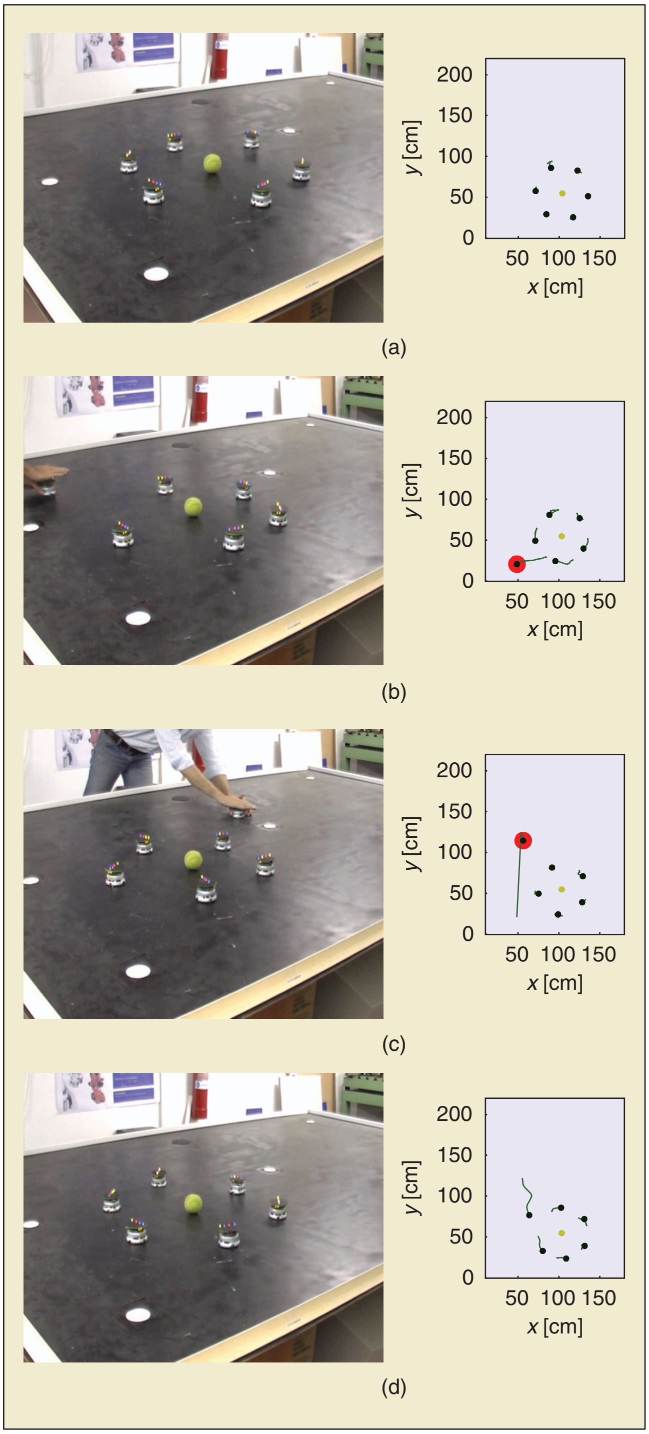


Figure 5. The second set of snapshots of the escorting experiment: from t ≈ 6 s to t ≈ 14 s. A fault is caused by (b) moving away a robot by hand and (c) further obscuring it to the camera. (d) After the reconfiguration is successfully achieved, the robot is put again in the arena.

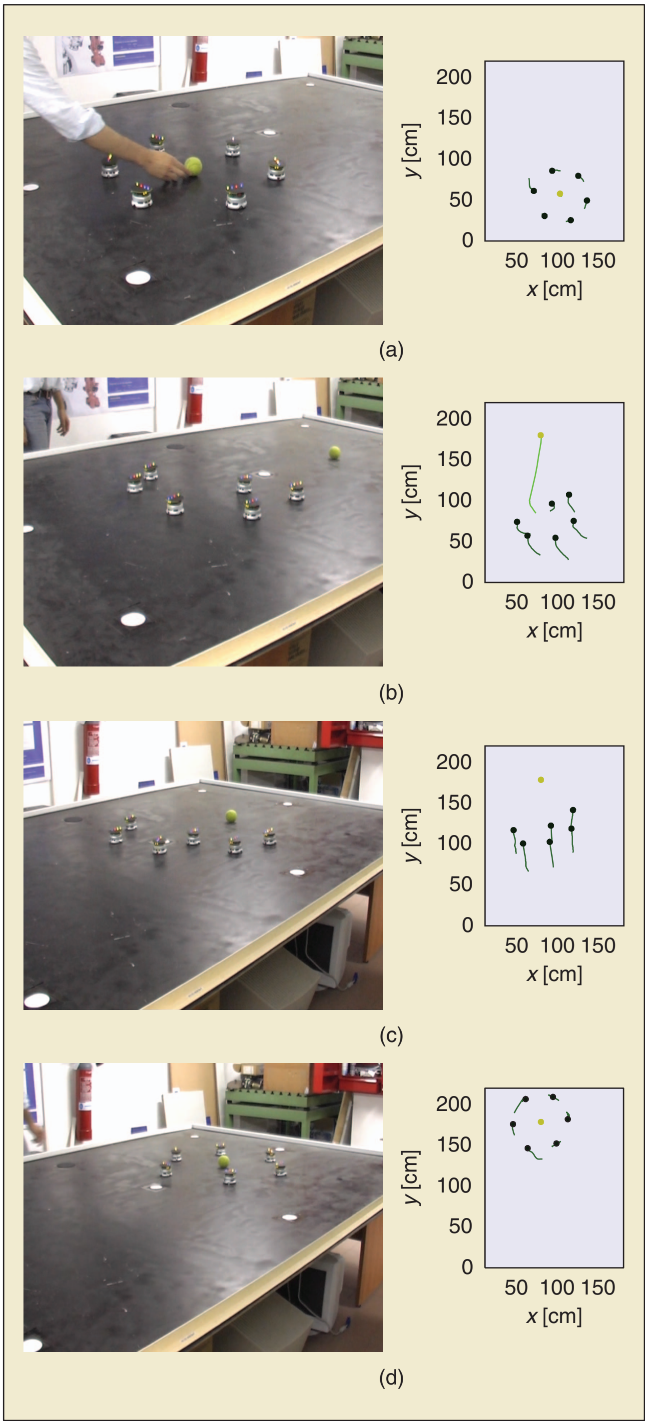


Figure 6. Third set of snapshots of the escorting experiment: from t ≈ 14 s to t ≈ 29 s.

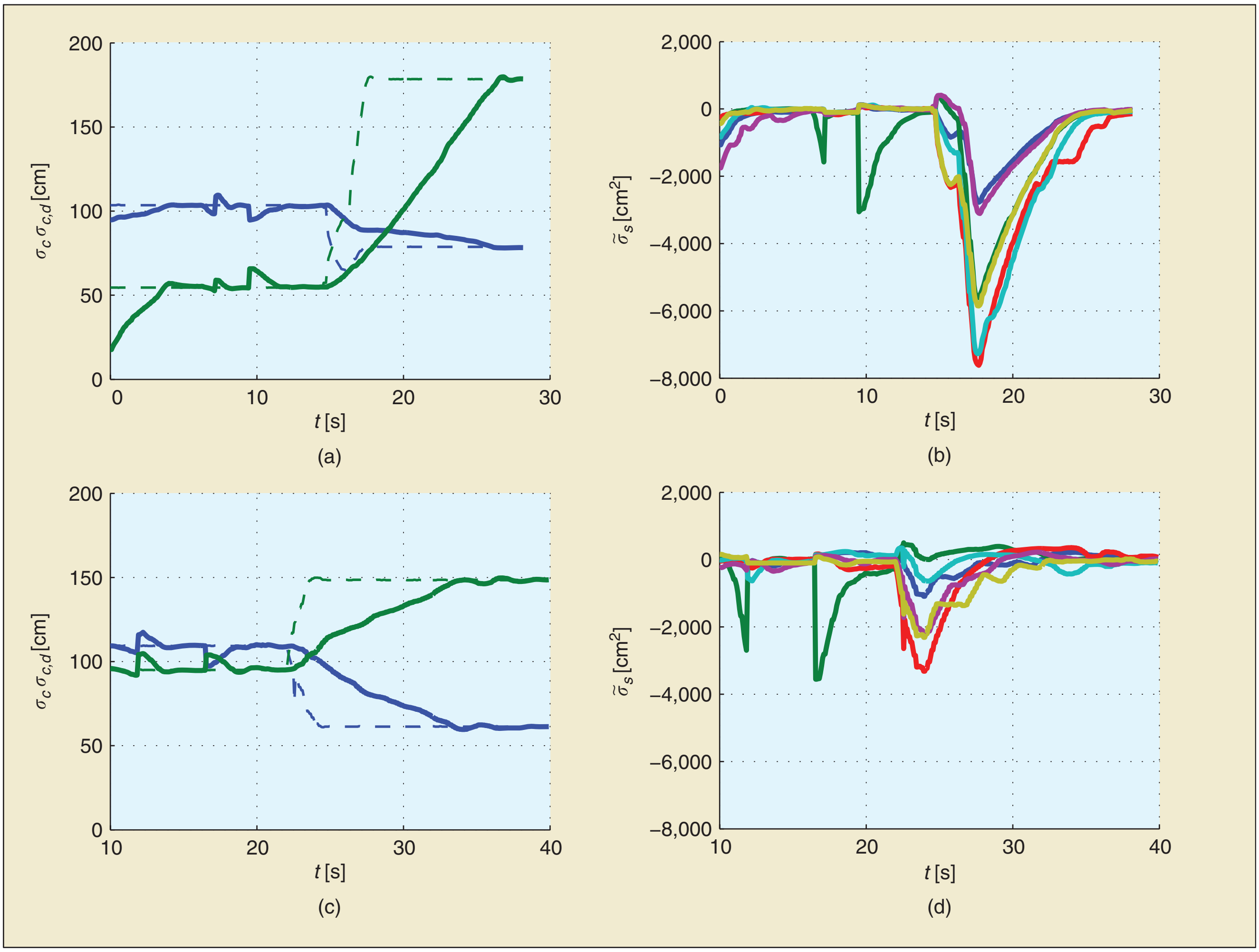


Figure 7. *Errors of the (a)–(c) centroid and circular (c)–(d) task functions relative to the two experiments shown with different tasks' priority order. (a)–(b) obstacle + circular + centroid + polygon and (c)–(d) obstacle + centroid + circular + polygon.*

in Figure 6, where the last ≈15 s of the mission is reported. Moreover, it can also be observed that, after the motion of the target, the vehicles reconfigure themselves with a different position relative to the first steady-state condition.

A second experiment was done that differs from the first one only in the priority orders of the tasks. These are reported in Table 2 together with the corresponding CLIK gains. The complete experimental results are not reported here, but see [25] for a video of the complete experiment (named `RAM_CENTROID.mpg`).

Finally, the time history of the centroid task function (solid line) against its desired value [i.e., the ball position (dashed line)] and the time history of the errors of the circular task function for both the experiments are reported in Figure 7. The errors are first convergent to zero. Then, several transients caused by the abrupt fault, the abrupt vehicle recovery, and the target movement can be observed. The behavior of the team in the two experiments is quite similar. However, it is worth noting from Figure 7(a) and (b) that the circular task function has a more regular shape when it has higher priority (in the first experiment).

Conclusions

The problem of escorting a moving target with a team of mobile robots was solved in this article by resorting to a formation control algorithm that can be cast in the framework of the NSB control approach. The overall mission, therefore, 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 validity of the proposed approach has been proved by both simulation case studies and experimental results with a team of six Khepera II mobile robots. Stability analysis concerning effective conditions needed to verify that the behaviors of specific missions are properly defined and merged is under investigation. Future improvements might regard decentralization of the algorithm, consideration of the vehicles' nonholonomicity in the definition of the task functions, and the introduction of a piecewise-constant constraint for the linear velocity to allow application of the method to teams of cruise vehicles (e.g., a fleet of vessels or a flight of planes).

Keywords

Multirobot, coordination control, behavioral approach.

References

[1] K-Team. [Online]. Available: http://www.k-team.com/
[2] 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. 2007 IEEE Int. Conf. Robotics and Automation*, Rome, 2007, pp. 1068–1073.

[3] 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, Jan. 2008.

[4] G. Antonelli and S. Chiaverini, "Kinematic control of a platoon of autonomous vehicles," in *Proc. 2003 IEEE Int. Conf. Robotics and Automation*, Taipei, 2003, pp. 1464–1469.

[5] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, pp. 1285–1292, Dec. 2006.

[6] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proc. IEEE*, vol. 94, no. 7, pp. 1306–1324, July 2006.

[7] B. Bishop and D. Stilwell, "On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles," in *Proc. 2001 IEEE Int. Conf. Control Applications*, México, 2001, pp. 823–828.

[8] E. Borhaug, A. Pavlov, R. Ghabcheloo, K. Pettersen, A. Pascoal, and C. Silvestre, "Formation control of underactuated marine vehicles with communication constraints," in *Proc. 7th IFAC Conf. Manoeuvring and Control of Marine Craft*, Lisbon, 2006.

[9] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE J. Robot.*, vol. 21, no. 3, pp. 376–386, June 2005.

[10] S. Carpin and L. Parker, "Cooperative motion coordination amidst dynamic obstacles," *Distrib. Autonom. Robot. Syst.*, vol. 5, pp. 145–154, 2002.

[11] L. Chaimowicz, B. Grocholsky, J. Keller, V. Kumar, and C. Taylor, "Experiments in multirobot air-ground coordination," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, p. 4053.

[12] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 13, no. 3, pp. 398–410, 1997.

[13] E. Fiorelli, N. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. Fratantoni, "Multi-AUV control and adaptive sampling in monterey bay," in *Proc. IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations*, Sebasco, ME, June 2004, pp. 134–147.

[14] A. Howard, L. Parker, and G. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *Int. J. Robot. Res.*, vol. 25, nos. 5–6, p. 431, 2006.

[15] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. Ganino, M. Garrett, S. Joshi, and P. Schenker, "CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration," *IEEE Trans. Syst. Man Cybern. A*, vol. 33, no. 5, pp. 550–559, 2003.

[16] I. Ihle, R. Skjetne, and T. Fossen, "Nonlinear formation control of marine craft with experimental results," in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, The Bahamas, 2004, vol. 1, pp. 680–685.

[17] T. Kamano, T. Yasuno, T. Suzuki, Y. Hasegawa, H. Harada, and Y. Kataoka, "Design and implementation of fuzzy cooperative catching controller for multiple mobile robots," in *Proc. 26th Ann. Conf. of the IEEE Industrial Electronics Society*, 2000, pp. 1749–1754.

[18] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams," *Experimental Robotics IX*. (Springer Tracts in Advanced Robotics), New York: Springer Verlag, 2005.

[19] M. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robot. Autonom. Syst.*, vol. 16, no. 2, pp. 321–331, 1995.

[20] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[21] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: design, implementation, and experimental validation," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 6, pp. 835–852, 2002.

[22] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, no. 1, p. 3, 2002.

[23] P. Pirjanian and M. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, 2000, vol. 3, pp. 2696–2702.

[24] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 3, pp. 201–212, 1990.

[25] Robotics Research group at the Industrial Automation Laboratory, University of Cassino, Cassino, Italy. [Online]. Available: http://webuser.unicas.it/lai/robotica/video/

Gianluca Antonelli received the laurea degree in electronic engineering and the research doctorate degree in electronic engineering and computer science from the University of Naples, Italy, in 1995 and 2000, respectively. He is currently an associate professor at the University of Cassino, Italy. He is a Senior Member of the IEEE since June 2006. Since September 2005, he has been the associate editor of *IEEE Transactions on Robotics*. He is also the editor of the *Journal of Intelligent Service Robotics* since November 2007. His research interests include simulation and control of underwater robotic systems, force/motion control of robot manipulators, multirobot systems, and identification. He has published 22 international journal papers and about 60 conference papers. He is the author of *Underwater Robots* (Springer Verlag; first edition, 2003; second edition, 2006).

Filippo Arrichiello received the laurea degree in mechanical engineering from the University of Naples, Italy, and the research doctorate degree in electrical and information engineering from the University of Cassino, Italy, in 2003 and 2007 respectively. He is currently a research assistant at the University of Cassino, Italy. From March to September 2005, he attended the Marie Curie Program at the Norwegian University of Science and Technology in Trondheim, Norway. His research activity mainly concerns industrial and mobile robotics with specific interest in multirobot system applications. He has published 14 journal and conference papers.

Stefano Chiaverini received the laurea and the research doctorate degrees in electronics engineering from the University of Naples, Italy, in 1986 and 1990, respectively. He is currently a professor of automatic control at the engineering faculty of the University of Cassino, Italy, where he is vice head of the Department of Automation, Electromagnetics, Information Engineering, and Industrial Mathematics. His research interests include manipulator inverse kinematics techniques, redundant manipulator control, cooperative robot systems, force/position control of manipulators, underwater robotic systems, and mobile robotic systems. He has published more than 150 international journal and conference papers, and he is the coeditor of *Complex Robotic Systems* (Springer Verlag, 1998) and is the coauthor of *Fondamenti di Sistemi Dinamici* (McGraw Hill, 2003). From January 2000 to February 2004, he has been an associate editor of *IEEE Transactions on Robotics and Automation* and, since August 2004, the editor of *IEEE/ASME Transactions on Mechatronics*. He is a Senior Member of the IEEE and a member of the IEEE Robotics and Automation Society conference editorial board.

Address for Correspondence: Filippo Arrichiello, Università degli Studi di Cassino, Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Via G. Di Biasio 43, 03043, Cassino (FR), Italy. E-mail: f.arrichiello@unicas.it.