PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/s13230-010-0006-0 · JBR · 1(1) · 2010 · 48-56

The NSB control: a behavior-based approach for multi-robot systems

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

Received 24 February 2010

Accepted 19 March 2010

Abstract

The paper presents an overview on the use of a behavior-based approach, namely the Null-Space-based Behavioral (NSB) approach, to control multi-robot systems in a wide application domain. The NSB approach has been recently developed to control the motion of generic robotic systems; it uses a projection mechanism to combine the multiple, prioritized, behaviors that compose the robotic mission so that the lower priority behaviors do not effect the higher priority ones. In this paper we describe how the NSB approach has been used to control different multi-robot systems (e.g., composed of wheeled and marine robots) to achieve missions such as formation control, entrapping/escorting of targets, control of mobile ad-hoc networks, flocking, border patrol and cooperative caging.

Keywords

multi-robot · behavior-based approach · mobile robots · marine robots

1. Introduction

Behavior-based robotics has been the object of wide research interest in the last decades, and, nowadays a consolidated literature on the field exists; for example, the textbook [1] offers a comprehensive state of the art on the field. Behavior-based approaches are methodologies to design the control architecture of artificial intelligence systems. The key idea of behavior-robotics is that the intelligence of the robotic system is provided by a set of behaviors, designed to achieve specific goals, that are activated on the basis of sensor information. Among the behavioral approaches, seminal works are reported in the papers [2] and [3]. In [2], the so-called layered architecture is proposed, where each behavior is related to a layer that is an asynchronous module that communicates over a low-bandwidth channel. On the basis of sensor information, each layer, working independently from the others, elaborates an output that is a motion command to the robot. Layers have different priority levels, and the possible conflict among the behaviors is solved by assigning a hierarchy so that the higher-level behaviors can subsume the lower-level ones. This architecture, also known in literature as ***subsumption architecture***, needs the use of a priority-based coordination function, and it is an example of competitive methods that provide a means of coordinating behavioral response for conflict resolution. In [3], the ***motor schema control*** is presented; that is a cooperative method where a behavioral fusion provides the ability to concurrently use the output of more than one behavior at a time. A supervisor elaborates each behavior, and it gives as output an intermediate solution, calculated as the sum of all the motion commands (one for each behavior) opportunely scaled by a gain vector. The supervisor, on the basis of sensor information, can dynamically change the gain vectors, giving instantaneously more or less weight to each behavior.

A behavioral approach designed for exploration of planetary surfaces has been investigated in [4], while in [5] the experimental case of an off-road navigation is presented. In [6] a hierarchical behavior-based system that performs several vision-based manipulation tasks by using different combinations of the same set of basic behaviors is presented. In [7] an architecture for dynamic changes of the behavior selection strategies is presented.

The behavior-based approach has also been useful for robotic researchers to examine the social characteristics of insects and animals, and to apply these findings to the design of multi-robot systems. 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, dispersing) in cooperative robotic systems. The first works were motivated by computer graphic applications; in 1986 Reynolds [8] made a computer model for coordinating the motion of animals as bird flocks or fish schools. This pioneer work inspired significant efforts in the study of group behaviors [9–11], and then in the study of multi-robot formations [12–15].

In the last years, we proposed a new behavior-based approach, namely the Null-Space-based Behavioral (NSB) control, that differs from the main approaches of this category in the behavioral coordination, that is in the way the behaviors are merged to define the final motion directives to the robots. In particular, the behaviors are arranged in priorities, and they are composed using null-space projection matrices so that multiple behaviors are simultaneously activated but the lower-priority behaviors do not affect the higher-priority ones. In fact, the NSB control always fulfils the highest-priority task; the lower-priority tasks, on the other hand, are fulfilled only in a subspace where they do not conflict with the ones having higher priority. This is clearly an advantage with respect to the competitive approaches, where one single task can be achieved at once, and to the cooperative approaches, where the use of a linear combination of each single task's output has as a result that

*E-mail: antonelli,f.arrichiello,chiaverini@unicas.it
†http://webuser.unicas.it/lai/robotica

no single task is exactly fulfilled. Moreover, differently from the classical behavior-based approaches, the analytical structure of the NSB approach allows to elaborate stability properties of the robotic mission. In [16] the NSB application to the control of generic robotic systems has been presented. Then, the NSB strategy has been applied to the control of different robotic systems composed of single and multiple mobile robots. In [17–20] we presented how the NSB technique can be used to control wheeled multi-robot systems to execute missions like formation control with obstacle avoidance, escorting/entrapping of an external target, control of robotic mobile ad-hoc networks, and flocking. In [21] and, successively, in [22, 23] we presented the extension of the NSB approach to the control of fleets of marine vehicles to execute missions, like formation control in the presence of current, cooperative target visiting with communication constraints, and cooperative caging of floating objects. In [24], the NSB application to the case of robotic systems with velocity saturation actuators is presented. In this paper we want to make a survey on the possible uses of the NSB approach to control multi-robot systems. Thus, in the following sections, we firstly recall its mathematical principles, then we discuss design issues related to the proposed technique, and finally we discuss several missions executed with different multi-robot systems.

2. The NSB

Specifically, by defining as $\boldsymbol{\sigma} \in \mathrm{R}^{m\times 1}$ the task variable to be controlled for the specific behavior, and as $\boldsymbol{p} \in \mathrm{R}^{s\times 1}$ the system configuration, then:

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

with the corresponding differential relationship:

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

where m is the task function dimension, $\boldsymbol{J} \in \mathrm{R}^{m\times s}$ is the configuration-dependent task Jacobian matrix, and $\boldsymbol{v} := \dot{\boldsymbol{p}} \in \mathrm{R}^{s\times 1}$ is the system velocity. Notice that, in case of a team of l planar robots where $\boldsymbol{p}_i \in \mathrm{R}^{2\times 1}$ is the position of the i^{th} robot, then $\boldsymbol{p} = [\boldsymbol{p}_1^T \ldots \boldsymbol{p}_l^T]^T$, that makes $s = 2\,l$. Also notice that the only case of interest is $m \leq s$; otherwise if $m > s$, the task would either be unfeasible or the null space of a full rank $\boldsymbol{J}(\boldsymbol{p})$ (i.e. $\mathrm{rank}(\boldsymbol{J}(\boldsymbol{p})) = s$) would be empty thus preventing the possibility of controlling any other task.
Consider a generic behavior k defined by the task variable $\boldsymbol{\sigma}_k$ having a desired value $\boldsymbol{\sigma}_{d,k}$ and a Jacobian $\boldsymbol{J}_k$: the velocity reference for the system is computed starting from the desired values $\boldsymbol{\sigma}_{d,k}$ by solving the inverse kinematic problem at a differential level. In particular, being $m \leq s$ it is possible to make use of the Moore-Penrose pseudo-inverse Jacobian of the task function to choose the minimum-norm velocity that fulfils the task. Thus, the velocity reference of the generic k^{th} task can be calculated as

$$\boldsymbol{v}_k = \boldsymbol{J}_k^{\dagger} \left(\dot{\boldsymbol{\sigma}}_{d,k} + \boldsymbol{\Lambda}_k \widetilde{\boldsymbol{\sigma}}_k \right), \tag{3}$$

where $\boldsymbol{J}_k^{\dagger} = \boldsymbol{J}_k^{\mathrm{T}} \left(\boldsymbol{J}_k \boldsymbol{J}_k^{\mathrm{T}} \right)^{-1}$ (when $m \leq s$ and $\mathrm{rank}(\boldsymbol{J}_k(\boldsymbol{p})) = m$), $\boldsymbol{\Lambda}_k$ is a suitable constant positive-definite matrix of gains and $\widetilde{\boldsymbol{\sigma}}_k$ is the task error defined as $\widetilde{\boldsymbol{\sigma}}_k = \boldsymbol{\sigma}_{d,k} - \boldsymbol{\sigma}_k$.
It is worth noticing that the term $\boldsymbol{\Lambda}_k \widetilde{\boldsymbol{\sigma}}_k$ is added to counteract the numerical drift due to discrete-time integration. A detailed discussion of the issues related to the case of rank deficient Jacobian (i.e. $\mathrm{rank}(\boldsymbol{J}_k(\boldsymbol{p})) < m$) goes beyond the scope of this paper. However, it can be noticed that the right-inverse $\boldsymbol{J}_k^{\dagger}$ in equation (3) can be computed through singularity robust techniques as, by example, the SVD. Notice that by replacing the command (3) in the corresponding dynamic equation (2), the closed loop task error dynamics result in the exponentially stable equation

$$\dot{\widetilde{\boldsymbol{\sigma}}}_k + \boldsymbol{\Lambda}_k \widetilde{\boldsymbol{\sigma}}_k = \boldsymbol{0}. \tag{4}$$

When the mission is composed of multiple behaviors, the overall system velocity is obtained by properly merging the outputs of the individual behaviors. A velocity vector for each behavior is computed as if it was acting alone; then, before adding the single contribution to the overall vehicle velocity, a lower-priority behavior is projected onto the null space of the higher-priority behaviors so as to remove those velocity components that would conflict with it. If the subscript k of Equation (3) also denotes the task priority (with task 1 being the highest-priority one) the overall robot velocity is derived as:

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \boldsymbol{N}_{1,1} \boldsymbol{v}_2 + \boldsymbol{N}_{1,2} \boldsymbol{v}_3, \tag{5}$$

where $\boldsymbol{N}_{1,k}$ is the projection matrix into the null-space of the tasks from 1 to k (see Figure 1). In particular, defining $\boldsymbol{J}_{1,k}$ as

$$\boldsymbol{J}_{1,k} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_k \end{bmatrix}, \tag{6}$$

the null-space projection matrix $\boldsymbol{N}_{1,k}$ is elaborated as

$$\boldsymbol{N}_{1,k} = \left(\boldsymbol{I} - \boldsymbol{J}_{1,k}^{\dagger} \boldsymbol{J}_{1,k} \right). \tag{7}$$

Therefore, the generalization of equation (5) can be written in the form:

$$\boldsymbol{v}_d = \sum_{k=1}^{N_{task}} \bar{\boldsymbol{v}}_k \tag{8}$$

with

$$\bar{\boldsymbol{v}}_k = \begin{cases} \boldsymbol{v}_1 & \text{if } k = 1 \\ \boldsymbol{N}_{1,k-1} \boldsymbol{v}_k & \text{if } k > 1. \end{cases}$$

A detailed convergence and stability analysis of the closed loop task error dynamics is reported in [25] where the concepts of orthogonality and independency are introduced. In the cited paper, it is also shown that the solution given by equations (6-8) gives rise to stable and convergent task error dynamics under very mild conditions on the task Jacobians $\boldsymbol{J}_k$ and $\boldsymbol{J}_{1,k}$. Moreover, under stronger assumptions related to the orthogonality and independency of the task Jacobians involved, also the solution

$$\bar{\boldsymbol{v}}_k = \begin{cases} \boldsymbol{v}_1 & \text{if } k = 1 \\ \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger} \boldsymbol{J}_1 \right) \ldots \left(\boldsymbol{I} - \boldsymbol{J}_{k-1}^{\dagger} \boldsymbol{J}_{k-1} \right) \boldsymbol{v}_k & \text{if } k > 1. \end{cases} \tag{9}$$

gives rise to convergent and stable closed loop task error dynamics. This solution can be represented via the block schema in Figure 2.

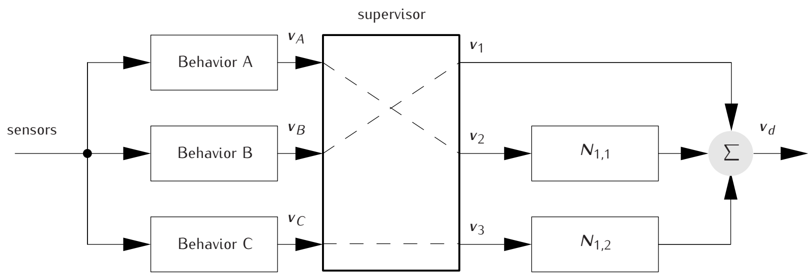

Figure 1. Sketch of the Null-Space-based Behavioral control in a 3-behavior example given by equation 5. The supervisor is in charge of changing the relative priority among the behaviors.

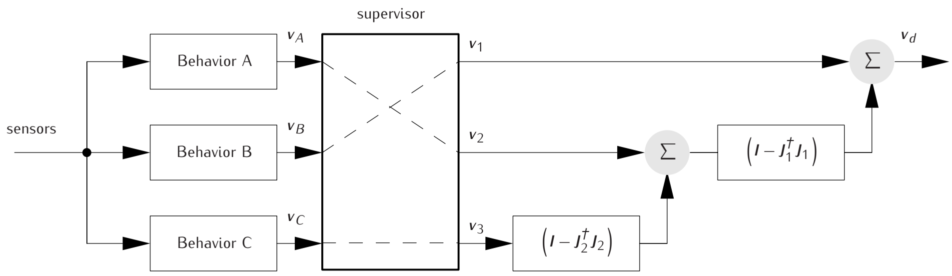

Figure 2. Sketch of the Null-Space-based Behavioral control in a 3-behavior example given by equation 9.

3. How to design elementary behaviors

To elaborate the velocity commands to the robots, the NSB control merges behaviors that have been defined in advance and arranged in priority. However, the design choices concerning how to define the elementary behaviors to achieve the assigned mission, and how to organize them in priority should be further discussed. These choices derive from practical considerations related to both the mission objective and the hardware/software characteristics of the robotic system.
For example, if the mission objective is to move the team of robots toward a desired area, we can define a behavior that controls the mean position of the robotic system. This behavior can be analytically described through a task function that elaborates the centroid position of the team; in a 2-dimensional 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. Thus, assigning a desired value to this task, we can compute the velocity commands to the robots elaborated on the base of Equation 3, and move the platoon toward the desired location.
As a further example, we can define a behavior to make the robots avoid collisions with obstacles or neighboring robots; this behavior may control the robot's distance from obstacles to keep it above a desired safety value.
In a general case, the way the behaviors are defined depends on the mission design approach. In a typical top-down approach, the roboticist has an overall idea of the mission to assign to the robotic system, and he decomposes the overall mission in elementary sub-problems. For each of them, he defines an elementary behavior and describes it through a mathematical task function. Then, he defines a priority order depending on practical consideration (e.g., safety behaviors as obstacle avoidance have always high priority) or on design choice on which behavior, in the case of conflict, needs to be achieved. In a bottom-up design approach, instead, the roboticist may start by defining a set of elementary behaviors and arranging them in priority. The global behavior of the system thus emerges as a composition of local behaviors. A bottom-up approach proves useful when the NSB is used to model and simulate dynamical systems as, for example, traffic control systems, biological systems, and swarms of robots. However, our focus is on how to define the behaviors when the mission objective are preventively well known. Thus, we mainly use top-down design approaches.

The behavior that controls the centroid positioning of the robots is described through a global task function, that is, a function that needs information about the absolute positions of all the robots of the team. Such a task function can be easily elaborated in the presence of a centralized system that collects information about all the robots. In an indoor environment, a global positioning estimation can be achieved via a ceiling vision system that sees all the robots. In an outdoor scenario, the robots can use their absolute localization system (as the GPS) and send this information to the central unit via the communication network. Once all the information has been collected, the central unit elaborates the desired velocity for each robot on the basis of Equation 3.

Although global behaviors can be useful to achieve missions where we need global control of the team positioning (as formation control or entrapping/escorting of a target), their usage is not allowed for distributed robotic systems where each robot has only access to local information. In the latter case, only decentralized behaviors can be used; that is,

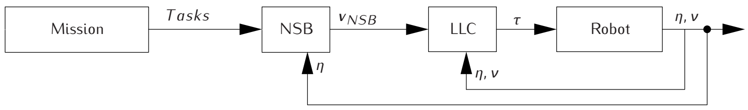

Figure 3. *Control architecture of the NSB with Low-Level Control.*

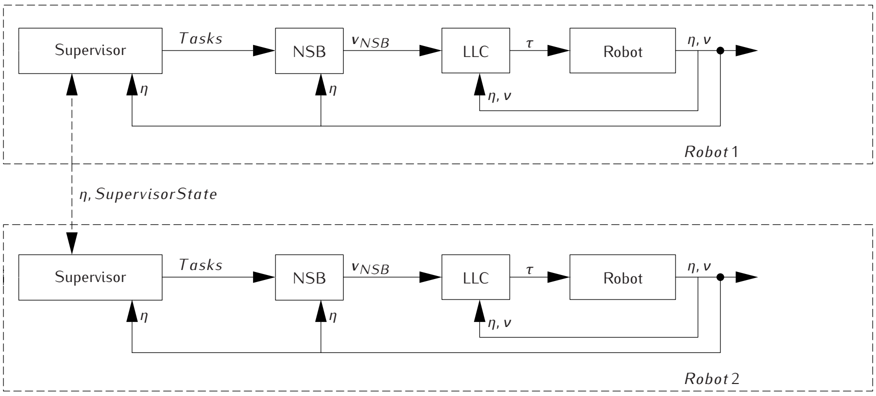

Figure 4. *Sketch of the control architecture with two cooperative robots.*

behaviors defined for an individual robot, and that only uses local information concerning the robot's neighbors (e.g., the relative positioning or the distance from its neighbors) or information provided by its local sensors. In this case, the behavior of the team emerges as a composition of local behaviors, similarly to the case of the bottom-up design approach; however, in this case the roboticist is required to design local behaviors that, individually implemented on board each robot, make the overall team perform as request for the specific mission.

4. Control architecture for different robotic systems

In this section we want to discuss how the NSB can be used with different robotic systems, e.g., wheeled, marine or flying robots. It is worth noticing that the NSB performs all the computations considering the robots as material points with first order dynamics; thus, its output is a linear velocity vector for each one of the robots. Depending on the robot kinematical and dynamical characteristics, this velocity vector may or may not be instantaneously achieved by the vehicle. For example, a non-holonomic robot (e.g., with unicycle-like or car-like kinematics) cannot instantaneously move in all the directions. In fact, it can move in the forward direction and rotate, but it cannot move in the lateral direction. Thus, when a desired velocity vector is assigned to it, the robot can not realize the desired motion if its orientation is not aligned with the desired velocity vector, but a proper maneuver that modifies its orientation needs to be performed. To this aim, a Low-Level Control (LLC) is designed for the specific kinematic/dynamic structure of the robot to make it turn and follow the linear velocity vector received by the NSB. As shown in Figure 3, the output vector of the NSB is given as a reference value to the LLC that, considering the kinematical and dynamical characteristics of the robots, is demanded to define the commands to the actuators (e.g. wheels velocities in the case of grounded robots, thrusters command in the case of marine robots). This two-level control loop organization allows the usage of the NSB strategy with different kinds of robotic systems neglecting their kinematical and dynamical characteristics (e.g., omnidirectional/non-holonomic wheeled robots, or fully-actuated/under-actuated marine robots). These aspects, instead, are considered by the LLC that is properly designed for specific robotic system.

Depending on the complexity of the mission, the set of active behaviors may change during the course of the mission. In this case, a supervisor module is also designed to decide, depending on the mission stage and using information about the environment, the active behaviors, their priorities and their reference values. For example, the supervisor can be organized as a finite state machine where, for each state, a set of behaviors is defined and their reference values are elaborated. This information is then given to the NSB module. Moreover, when the robots of a distributed system are commanded to execute a cooperative mission, they may also need to communicate with their neighbors; in this case, the supervisor is also in charge of managing the communication. A sketch of the described control architecture for a two-robot system is reported in Figure 4.

5. Experimental missions

The NSB approach has been used to control multi-robot systems in several scenarios. In the following we briefly discuss a set of performed missions. However, for major details on the single missions, the readers are redirected to the specific papers referenced in the specific sections. Videos of mission executions with real multi-robot systems can

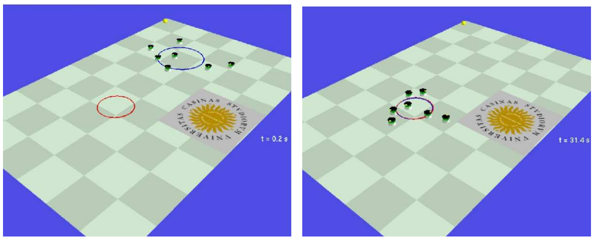

Figure 5. *Reconstruction of a spread control mission execution with a team of mobile robots. The two snapshots represent the initial (left) and final (right) robots configuration, where the circles represent the measured and desired centroid position and variance.*

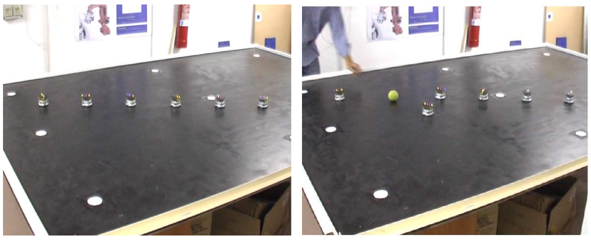

Figure 6. *Two snapshots of a formation control mission with a moving obstacle passing through the formation.*

be found at the URL: http://webuser.unicas.it/lai/robotica

5.1. Centralized missions

We firstly discuss a set of missions that make use of centralized behaviors. The following experiments have been performed using a centralized unit that, receiving information about the robot positioning from a vision system, elaborates the NSB algorithm and sends the motion commands to the robots via Bluetooth communication.

5.1.1. Spread control

The first mission we consider concerns the possibility to control the positioning of a team of robots and their spread in the environment; that is, we want to assign the mean positioning of the team and how much the robots have to stay close/far from the others. The mission is decomposed into three elementary behaviors that are:

1. collision avoidance;
2. centroid;
3. variance.

The first behavior is aimed at avoiding collisions among robots or with static obstacles eventually present in the environment. Its task function controls, for each robot, the distance from the closest obstacle or vehicle and keeps it above a threshold value. The centroid (or barycenter) task function is designed to control the mean positioning to the robots so that, using a single parameter (the desired centroid positioning), we can move the overall team to an assigned area. Finally, we control the variance of the robots' displacement. By assigning its desired value we command the robots to stay close (with a low desired variance value) or largely spread in the environment (with high desired variance value). These behaviors are arranged in priority keeping the obstacle avoidance as the higher priority task since it is of crucial importance for the robotic team safety.
Since both the centroid and variance task functions require the knowledge of the positioning of the overall team, the task functions results centralized. In Figure 5, we report a reconstruction from experimental data of a mission execution performed with a team of mobile robots controlled via a centralized control unit. In the figure, the circles are centered in the measured and desired centroid position, and their radii are proportional to the measured and desired variance values. More details on task function definitions and on the mission execution can be found in [17].

5.1.2. Formation control

As a second mission, we want to control the exact relative and global positioning of the robots of the team, thus we want to execute a so called formation control mission. For this mission, the designed elementary behaviors are:

1. collision avoidance;
2. centroid;
3. rigid formation.

The first two behaviors are the same for the spread control mission. The rigid formation, instead, expresses the relative positioning of each robot with respect to the centroid. Assigning its desired values we define the desired formation geometry in which we want to arrange the robot. Keeping the centroid and rigid formation decoupled, we can individually control the mean position of the team and their relative displacements by assigning respectively a desired centroid positioning and a desired geometric formation. In Figure 6 two snapshots of a formation control mission where the robots were commanded to reach a linear formation, with a fixed centroid positioning, are reported. Since the primary task is the collision avoidance, when an external obstacle is close to the formation (as in the right plot of Figure 6) the robots have to give priority to the obstacle avoidance and temporarily lose the formation. Then, when the obstacle has moved far away, the assigned formation is reached again. From the video related to this mission, where a moving obstacle (a tennis ball pushed by hand) is thrown over the moving team of robots, it is easy to notice the described behavior.
In the formation control mission, the desired geometric formation is not requested to be static. In the mission whose sketch is reported in Figure 7, the robots are requested to reach a circular formation and, as soon as the formation is reached, the desired configuration switches to a new one. In particular, always keeping a circular displacement, each robot is requested to exchange its position with the opposite robot in the circle. In such a way, as soon as the new formation is commanded, all the robots converge toward the center of the circle creating a high-traffic condition. Such mission has been studied to test the collision avoidance function in a cluttered environment. Further details on these two mission executions can be found in [17].

5.1.3. Target escorting

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 multi-robot system has to entrap the target and reduce its possible escape windows

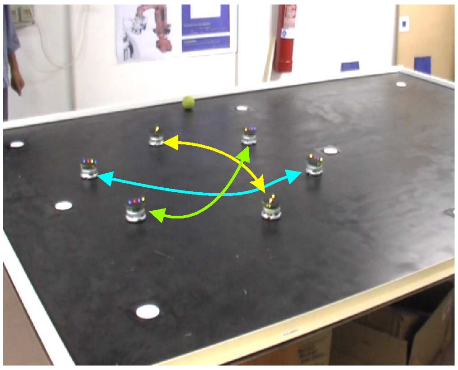

Figure 7. *Circular permuting formation.*

Figure 8. *The entrapment/escorting mission.*

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 on the target and whose sides define a sort of intrusion/escape window (see Figure 8).
Following the NSB approach, the escorting mission can be executed through the following elementary behaviors:

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

The execution of the entrapping/escorting mission with a team of six mobile robots has been presented in [18]. The approach does work also in the case of robots' faults. When one robot is out of work, the others are commanded to rearrange their displacement to guarantee the previously assigned escape windows, thus they rearrange the radius of the circumference depending on the number of active robots.

5.2. Distributed missions

In the previous section we have recalled missions where the use of centralized behavior is allowed. However, such behaviors cannot be used to control distributed robotics systems where each robot may only use local information. In this case, the NSB algorithm runs on board of each robot independently from the controller on board of the other vehicles. Thus, we need to identify behaviors for single robots that make only use of local information concerning the robot's neighbors or close obstacles. The global behavior of the team emerges as a consequence of the local behaviors of the single robots, thus we have to define local behaviors that, implemented on board the single robots, make the overall team behave as requested by the mission objectives. The experiments described in the following of this section have been performed with different typologies of multi-robot systems. Flocking and MANET missions have been experimentally tested using a centralized system that simulates a decentralized structure; that is, the motion directives to each robot are elaborated by the central unit making only use of local information relative to the individual robot (i.e., the distances from its neighbors). The border patrol and the mission with marine robots, instead, have been performed with decentralized multi-robot systems.

5.2.1. Flocking

Flocking is an example of distributed mission (see [20]). In this case, the mission consists of making a swarm of robots group together into a $lattice$ configuration, that is a geometric configuration characterized by the fact that all the robots have the same distances from their neighbors. The global emerging behavior is obtained by implementing individual controllers onboard each robot. The robots can only sense their relative positions with respect to their neighbors; moreover, when the robot team has to converge toward a rendez vous point, each robot also needs its absolute position.

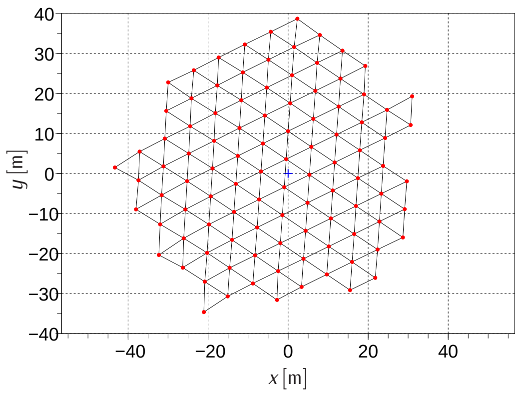

Figure 9. 100 *robots flocking with a rendez vous point.*

Each of the robots is only aware of the robots inside its sensing range. To decide the active behaviors, each i^{th}-robot lists the neighboring robots in a vector $\boldsymbol{k}_i$ sorted on the base of their relative distance from it (with $\boldsymbol{k}_i(1)$ being the closest neighbor). Referring to a 2-dimensional case, each robot computes the desired velocities corresponding to the following behaviors:

- Lattice behavior with respect to the robot $\boldsymbol{k}_i(1)$ (if there is one robot in $\mathcal{N}_i$);

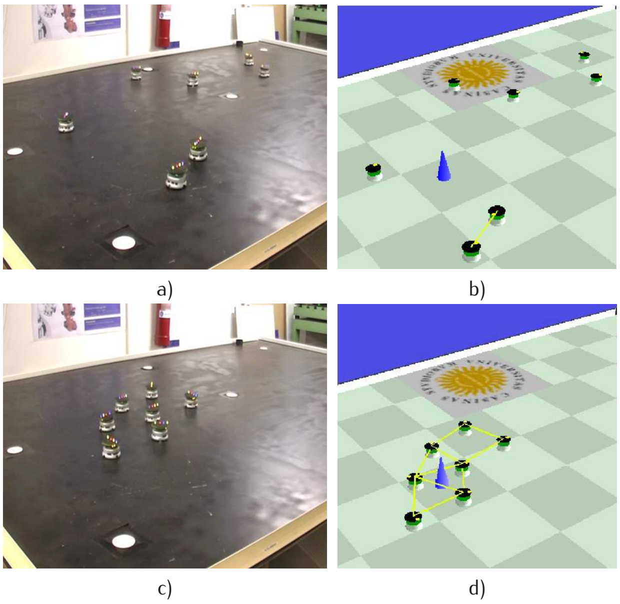

Figure 10. *Snapshots of the initial and final configurations and their graphical elaborations of an experiment where seven robots flock around a fixed rendez vous point.*

- Lattice behavior with respect to the robot $\boldsymbol{k}_i(2)$ (if there are two robots in $\mathcal{N}_i$);
- Moving towards rendez vous behavior;
- Obstacle avoidance behavior;

where the lattice behavior controls the distance from one specific robot, and moving towards the rendez vous pushes the robot towards an assigned location.

5.2.2. MANET

In the following, we consider a mission where a platoon of mobile antennas must ensure the communication between a mobile agent executing its mission and a fixed base station (e.g., an Internet access point). The antennas realize a Mobile Ad-hoc NETwork (MANET), and are commanded to ensure a multi-hop communication path between the agent and the base station. Moreover, the mobile agent and the antennas have to avoid the obstacles eventually present in the environment. The NSB algorithm is used to define the motion control strategy of the antennas to guarantee the mission objective. Each antenna is supposed to be able to communicate with the set of other antennas that dynamically are in its communication range, and, toward a multi-hop communication strategy, the antennas have to guarantee the communication between the agent and the base station. Assigned a virtual path from the agent to the base station in the communication network, each intermediate antenna in the path has a previous and a following antenna along the path. To ensure the global connectivity, each antenna is commanded to keep the connection with the previous and the following antennas through behaviors that control the intra-antennas distances. Thus, the behaviors to be achieved by the antennas in order of decreasing priority, 1 being the highest and assuming that k_0 is the base station, are:

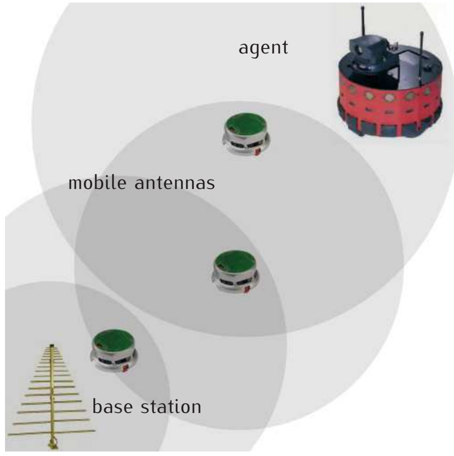

Figure 11. *Sketch of the coverage problem to be solved; the autonomous agent needs to be connected with the base station by the use of a platoon of mobile antennas.*

1. avoid the obstacles;
2. keep the next antenna k_{j+1} in the communication zone;
3. keep the previous antenna k_{j-1} in the communication zone.

Further details can be found in [19, 26].

5.2.3. Border patrol

The NSB approach has also been used to control a team of mobile robots patrolling an open or closed line [27]. The implemented control algorithm is fully decentralized, i.e., no communication occurs between robots or with a central station. Robots behave according only to their sensing and computing capabilities, to ensure high scalability and robustness to robots faults. The patrolling algorithm has been designed in the framework of NSB control and, in particular, it is based on the concept of Action: a higher level of abstraction with respect to the behaviors. Each Action is obtained by combining more elementary behaviors in the NSB framework. A supervisor, designed as a finite state machine, was in charge of selecting the appropriate action among the following set:

- Action Reach Frontier;
- Action Keep Going;
- Action Patrol Clockwise;
- Action Patrol Counter-Clockwise;
- Action Avoid Teammate.

Each action selects and prioritizes the active behaviors from the following set: reach frontier, patrol frontier clockwise, patrol frontier counter-clockwise, teammate avoidance. The approach has been validated in simulation as well as experimentally with a patrol of 3 Pioneer robots available at the Distributed Intelligence Laboratory of the University of Tennessee.

5.2.4. Missions with marine robots

Figure 12. Two autonomous surface vessels of the Robotic Embedded Systems Laboratory at USC. Courtesy of prof. Gaurav Sukhatme.

As a last case study, we propose the use of the NSB to control a collaborative team of marine robots. In [21] we have presented the usage of the NSB technique for the control of autonomous surface vessels executing a formation control mission in the presence of current and disturbances. Recently, two under-actuated autonomous surface vessels, property of the Robotic Embedded Systems Laboratory of the University of Southern California (see Figure 12), have been used to execute a cooperative navigation mission into the field while satisfying a communication constraint. In particular, the NSB technique has been tested in a mission where a set of target locations spread across a planar environment has to be visited once by either of the two ASVs while maintaining a relative separation less than a given maximum distance (to guarantee inter-vessels wireless communication). Experiments were carried out in the field with a team of two ASVs visiting 22 locations on a lake surface with static obstacles. Further details can be found in [22].
More recently, we have used the NSB approach to execute a caging mission on the water's surface. In particular, we have considered the problem of using two robotic boats, connected with a floating rope, to 'capture' a floating object from a known location on the water's surface and 'shepherd' it to a designated position. Preliminary results of the mission execution can be found in [23].

6. Conclusions

In this paper we present an overview on a behavior-based technique, namely the Null-Space-based Behavioral control, and its possible usages to control multi-robot systems executing coordinated missions. The scope of the paper is to give a tutorial on the use of the NSB approach, and to make an overview on the missions executed with real multi-robot systems. For each of the performed missions, the main issues and objectives are recalled, however, further details can be found in the referenced papers. The NSB approach has shown to be versatile and useful in a wide application domain and with different kind of multi-robot systems. Despite the approach has been already used in several scenarios, further missions, such as the border patrol and the caging mission, are still under investigation. Moreover, the NSB approach is going to be used to control teams of underwater robots to achieve operations in harbor scenarios. From a theoretical point of view, the NSB approach still presents some open issues, e.g. concerning how to execute centralized missions using a distributed system, or how to consider in the control design some specific limits of local sensors (as a limited view cone of an on-board camera); these aspects will be object of future studies. Finally, it is worth noticing that, despite the technique having been proposed quite recently, it is starting to have proselytes in the robotic community, and some research groups have started to independently use it for their research purposes, examples are the papers [28–30].

References

[1] R. C. Arkin. Behavior-Based Robotics. The MIT Press, Cambridge, MA, 1998.
[2] R. A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1):14–23, 1986.
[3] R. C. Arkin. Motor schema based mobile robot navigation. The International Journal of Robotics Research, 8(4):92–112, 1989.
[4] E. Gat, R. Desai, R. Ivlev, J. Loch, and D. P. Miller. Behavior control for robotic exploration of planetary surfaces. IEEE Transactions on Robotics and Automation, 10(4):490–503, 1994.
[5] D. Langer, J. K. Rosenblatt, and M. Hebert. A behavior-based system for off-road navigation. IEEE Transactions on Robotics and Automation, 10(6):776–783, 1994.
[6] A. Saffiotti and Z. Wasik. A hierarchical behavior-based approach to manipulation tasks. In Proceedings of 2003 IEEE International Conference on Robotics and Automation, pages 2780–2785, Taipei, Taiwan, Sept. 2003.
[7] M. Scheutz and V. Andronache. Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems. IEEE Transactions on Systems, Man and Cybernetics, 34(6):2377–2395, Dec. 2004.
[8] C. Reynolds. Flocks, herd and schools: A distributed behavioral model. Computer Graphics, 21(4):25–34, 1987.
[9] L. E. Parker. Designing control laws for cooperative agent teams. In Proceedings 1993 IEEE International Conference on Robotics and Automation, volume 3, pages 582–587, Atlanta, GA, May 1993.
[10] M. J. Matari¢. Designing emergent behaviors: From local interaction to collective intelligence. In Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animal to Animal, pages 432–441, 1992.
[11] C. R. Kube and H. Zhang. Collective robotics: from social insects to robots. Adaptive Behavior, 2(2):189–218, 1993.
[12] P. K. C. Wang. Navigation strategies for multiple autonomous robots moving in formation. Journal of Robotic Systems, 8(2):177–195, 1991.
[13] L.E. Parker. On the design of behavior-based multi-robot teams. Advanced Robotics, 10(6):547–578, 1996.
[14] M. J. Matari¢. Behavior-based control: Examples from navigation, learning, and group behavior. Journal of Experimental and Theoretical Artificial Intelligence, 9(2-3):323–336, 1997.
[15] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. IEEE Transactions on Robotics and Automation, 14(6):926–939, 1998.
[16] G. Antonelli, F. Arrichiello, and S. Chiaverini. The Null-Space-based Behavioral control for autonomous robotic systems. Journal of Intelligent Service Robotics, 1(1):27–39, Jan. 2008.
[17] G. Antonelli, F. Arrichiello, and S. Chiaverini. Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control. IEEE Transactions on Control Systems Technology, 17(5):1173–1182, Sept. 2009.
[18] G. Antonelli, F. Arrichiello, and S. Chiaverini. The entrapment/escorting mission: An experimental study using a multirobot system. IEEE Robotics and Automation Magazine (RAM). Special Issues on Design, Control, and Applications of Real-World Multi-Robot Systems, 15(1):22–29, March 2008.
[19] G. Antonelli, F. Arrichiello, S. Chiaverini, and R. Setola. Coordinated control of mobile antennas for ad-hoc networks. In-

ternational Journal of Modelling Identification and Control Special/Inaugural issue on Intelligent Robot Systems, 1(1):63–71, 2006.

[20] G. Antonelli, F. Arrichiello, and S. Chiaverini. Flocking for multi-robot systems via the Null-Space-based Behavioral control. Swarm Intelligence, 4(1):37–56, March 2010.

[21] F. Arrichiello, S. Chiaverini, and T. I. Fossen. Formation control of marine surface vessels using the Null-Space-based Behavioral control. In Group Coordination and Cooperative Control, K.Y. Pettersen, T. Gravdahl, and H. Nijmeijer (Eds.), Springer-Verlag's Lecture Notes in Control and Information Systems series, pages 1–19. May 2006.

[22] F. Arrichiello, J. Das, H. Heidarsson, A. Pereira, S. Chiaverini, and G. S. Sukhatme. Multi-robot collaboration with range-limited communication: Experiments with two underactuated ASVs. In Proceedings 2009 International Conference on Field and Service Robots, Cambridge, MA, USA, July 2009.

[23] F. Arrichiello, H. Heidarsson, S. Chiaverini, and G. S. Sukhatme. Cooperative caging using autonomous aquatic surface vehicles. In Accepted to 2010 IEEE International Conference on Robotics and Automation, 2010.

[24] F. Arrichiello, S. Chiaverini, G. Indiveri, and P. Pedone. The Null-Space based Behavioral control for mobile robots with velocity actuator saturations. to appear International Journal of Robotics Research, 2010.

[25] G. Antonelli, F. Arrichiello, and S. Chiaverini. Stability analysis for the Null-Space-based Behavioral control for multi-robot systems. In 47th IEEE Conference on Decision and Control and 8th European Control Conference, Cancun, MEX, Dec. 2008.

[26] G. Antonelli, F. Arrichiello, S. Chiaverini, S. Contraffatto, and R. Setola. Use of a robot platoon to implement mobile ad-hoc network in rescue scenario - preliminary experimental results. In IEEE International Workshop on Safety, Security, and Rescue Robotics, Rome, I, Sept. 2007.

[27] A. Marino, L. Parker, G. Antonelli, and F. Caccavale. Behavioral control for multi-robot perimeter patrol: A finite state automata approach. In Proceedings 2009 IEEE International Conference on Robotics and Automation, Kobe, J, May 2009.

[28] M. Cellini, R. Mati, L. Pollini, and M. Innocenti. Obstacle avoidance for autonomous ground vehicles in outdoor environments. In Intelligent Vehicles Symposium, 2007 IEEE, pages 258 –263, Istanbul, Turkey, June 2007.

[29] R. Schlanbusch, R. Kristiansen, and P. J. Nicklasson. Spacecraft formation reconfiguration with collision avoidance. In Proceedings of the 3rd International Symposium on Formation Flying, Missions and Technologies, Estec, Holland, 2008.

[30] V. Scordamaglia and M. Mattei. Null-space based behavioral approach control of an autonomous team of mobile robots. In 2nd Conf. on agro-industrial logistics in mediterranean area, Reggio Calabria, I, 2007.