Swarm Intell (2010) 4: 37–56
DOI 10.1007/s11721-009-0036-6

Flocking for multi-robot systems via the Null-Space-based Behavioral control

Gianluca Antonelli · Filippo Arrichiello · Stefano Chiaverini

Received: 4 December 2007 / Accepted: 23 September 2009 / Published online: 17 October 2009

Abstract Flocking is the way in which populations of animals like birds, fishes, and insects move together. In such cases, the global behavior of the team emerges as a consequence of local interactions among the neighboring members. This paper approaches the problem of letting a group of robots flock by resorting to a behavior-based control architecture, namely Null-Space-based Behavioral (NSB) control. Following such a control architecture, very simple behaviors for each robot are defined and properly arranged in priority in order to achieve the assigned mission. In particular, flocking is performed in a decentralized manner, that is, the behaviors of each robot only depend on local information concerning the robot's neighbors. In this paper, the flocking behavior is analyzed in a variety of conditions: with or without a moving rendez-vous point, in a two- or three-dimensional space and in presence of obstacles. Extensive simulations and experiments performed with a team of differential-drive mobile robots show the effectiveness of the proposed algorithm.

Keywords Flocking · Multiple mobile robots · Behavioral control

1 Introduction

In nature many living beings such as birds, fish, bacteria, and insects exhibit collective behaviors obtained by using local control strategies. Among the different behaviors, flocking

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s11721-009-0036-6) contains supplementary material, which is available to authorized users.

G. Antonelli (✉) · F. Arrichiello · S. 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
e-mail: antonelli@unicas.it
url: http://webuser.unicas.it/lai/robotica/

F. Arrichiello
e-mail: f.arrichiello@unicas.it

S. Chiaverini
e-mail: chiaverini@unicas.it

has fascinated researchers from several disciplines, e.g., physicists, social scientists, animal psychologists, and roboticists. As shown in Matarić (1995), the flocking problem is an interesting control problem involving the coordination of multiple robots characterized by limited sensing and communication capabilities, and it is strictly related to the study of self-organized networks of mobile robots. Thus, the flocking problem can be considered as a specific case in the study of the coordinated control of multiple robots including distributed sensing, search and rescue, exploration, and coverage. The work by Cao et al. (1997) provides a significant overview of these control problems.

In 1987, a seminal work by Reynolds (1987) presented a computer model for motion coordination of animals, such as bird flocks or fish schools. The aggregate motion of the flock team was the result of the interaction of relatively simple behaviors of the individual simulated birds, where each bird was simulated as an independent actor that navigated according to its local perception of the dynamic environment, the laws of simulated physics that rule its motion, and a set of programmed behaviors. An extensive literature now exists that reports interesting results concerning the flocking problem. In Olfati-Saber (2006), different solutions are investigated, and their stability analysis is discussed. Based on local sensing, each robot moves according to three different terms (a gradient-based term, a consensus term, and a navigational feedback term) that represent different *behaviors* of each robot. The work by Cortes et al. (2006) presents an algorithmic coordination approach for mobile agent networks to make the agents converge on a rendez-vous point without losing the connection with their neighbors. The work by Martinez et al. (2007) surveys recent developments in modeling, analysis, and design of distributed motion coordination algorithms for multi-robot systems. The work by Hsieh et al. (2008) presents a decentralized control strategy to make a team of robots converge to the boundary of regular shapes only using local interactions.

An aspect that strongly influences the coordination strategy is the possibility for the robots to explicitly exchange information with their neighbors; this possibility poses the challenging problem of *consensus*, that is, as shown in Olfati-Saber et al. (2007), reaching an agreement regarding a certain quantity of interest that depends on the state of all the agents. An overview of the information consensus is given in Ren et al. (2007) and Ren and Beard (2008), while the work Olfati-Saber et al. (2007) investigates consensus algorithms with emphasis on robustness, time-delays, and performance guarantee. The work by Ji and Egerstedt (2007) shows how a consensus variable can be used to make the flock perform particular behaviors like formation keeping or rendez-vous. In Tanner et al. (2007), the proposed decentralized controller is stable under arbitrary changes in the connected network. The work of Jadbabaie et al. (2003) presents a stability analysis of several decentralized strategies that achieve an emergent behavior. Nonholonomic agent motion is explicitly taken into account in Moshtagh and Jadbabaie (2007).

The study of autonomous robotics has been strongly influenced by the robotics paradigm of behavior-based control, introduced in the works of Brooks (1986) and Arkin (1989). Using sensors to obtain instantaneous information about the environment, the behavior-based approaches give the system the autonomy to navigate in complex environments. Thus, they can be used to control both single robots and multi-robot systems to navigate in unknown or dynamically changing environments. In this paper, a possible solution to the flocking problem is proposed by resorting to the behavioral approach defined as NSB (Null-Space-based Behavioral control) presented in Antonelli et al. (2008a). This approach, strongly related to the kinematic control presented in Antonelli and Chiaverini (2003, 2006), uses a hierarchy-based strategy to compose the elementary behaviors that constitute the overall mission of the multi-robot systems. In particular, in case of conflicting behaviors, the NSB deletes the motion components of the lower priority behaviors that would conflict with the higher-priority behaviors.

The NSB approach takes the advantages of behavior-based approaches, such as an easy control design and the reactivity to unknown or dynamically changing conditions; moreover, it presents a rigorous mathematical formulation that allows the proof of some analytical convergence properties (Antonelli et al. 2008b). However, unlike the behavioral approaches, it requires an analytical description of the behaviors. The NSB approach has been recently applied in a large number of experimental missions for multi-robot such as formation control and the escorting/entrapment of an autonomous target as shown in Antonelli et al. (2007, 2008c). Unlike these previous works, in this paper the NSB approach is used for the first time to achieve flocking; moreover, the paper presents the first use of the NSB as a decentralized control technique. Flocking, in fact, emerges as a global behavior obtained by using only local controllers for each single robot, that is, each robot requires only local information such as its relative position with respect to its neighbors and, only in the case of a rendez-vous, its global position; the control strategy can thus work in a totally decentralized manner. Extensive simulations, assuming two- or three-dimensional point-mass robots, and two-dimensional experiments using a platoon of seven differential-drive mobile robots, namely the Khepera II, show the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 presents the main flocking concepts and definitions. Section 3 introduces the basic concept of the Null-Space-based Behavioral control. Section 4 presents the behavior functions to achieve the flocking mission via the NSB approach and presents the supervisory control strategy. Section 5 presents the results of several numerical simulations to achieve flocking in different conditions. Section 6 presents the experimental results achieved with a team of seven real mobile robots. Finally, Sect. 7 presents some conclusions and suggests some directions for future works.

2 The flocking problem

The flocking problem has been approached by several researchers of different disciplines. For this reason, the word *flocking* assumes slightly different meanings in the literature. In this paper, flocking of a swarm of robots is considered as the aim of grouping them into a *lattice* configuration, where the *emerging* behavior is obtained by implementing individual controllers in each robot. The robots can only sense their relative positions with respect to their neighbors; moreover, when the robot team has to converge to a rendez-vous point, each robot also needs its absolute position.

Firstly, the basic notions of graph theory are briefly discussed. A graph G is a pair $(\mathcal{V}, \mathcal{E})$ that consists of a set of vertices $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$. In this paper, an undirected graph will be considered, thus, $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$. The scalar quantities $|\mathcal{V}|$ and $|\mathcal{E}|$ will be denoted as the order and the size of the graph, respectively. The adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ contains the information concerning the edges: $a_{i,j} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E}$; for an undirected graph, $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$. The set of neighbors of node i is defined as

$$\mathcal{N}_i = \{j \in \mathcal{V} : a_{i,j} \neq 0\} = \left\{j \in \mathcal{V} : (i, j) \in \mathcal{E}\right\}. \tag{1}$$

In our case, the graph is related to the position of nodes in the Euclidean space. Thus, we denote the position of each node as $\boldsymbol{p}_i \in \mathbb{R}^l$, where $l = 2$ in 2D-space, or $l = 3$ in 3D-space. Then, the configuration of all the nodes of the graph is represented by the vector $\boldsymbol{p} \in \mathbb{R}^{ln}$ defined as

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1^{\mathrm{T}} & \boldsymbol{p}_2^{\mathrm{T}} & \cdots & \boldsymbol{p}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}.$$

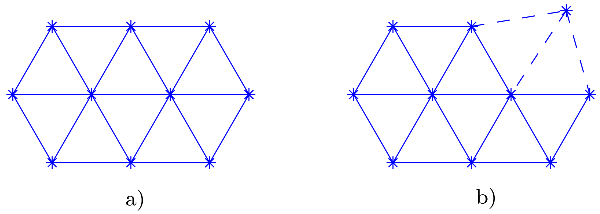


Fig. 1 Examples of α-Lattice and quasi-α-Lattice for 10 robots in a 2D space

A framework, or structure, is a pair $(G, \boldsymbol{p})$ that consists of a graph and the configuration of its nodes. In such a case, the graph includes an edge between two nodes ($a_{i,j} \neq 0$) if their Euclidean distance is smaller than a threshold value, called the *interaction range* r. Thus, (1) is equivalent to

$$\mathcal{N}_i = \{ j \in \mathcal{V} : \|\boldsymbol{p}_i - \boldsymbol{p}_j\| < r,\ j \neq i \}, \tag{2}$$

where $\|\cdot\|$ represents the Euclidean norm.

In order to describe the spatial order of the desired configuration of flocking in a proper analytical framework, the definition of the *α-Lattice* structure is introduced. The *α-Lattice* configuration represents a geometric structure that satisfies

$$\|\boldsymbol{p}_i - \boldsymbol{p}_j\| = d, \quad \forall j \in \mathcal{N}_i \text{ and } \forall i \in \mathcal{V}, \tag{3}$$

where d is the *lattice scale*. That is, an α-Lattice is a geometric configuration characterized by the fact that all the edges of the graph have the same length (see Fig. 1a). Moreover, the *lattice ratio* κ is defined as the ratio between the interaction range and the lattice scale, that is, $\kappa = r/d$.

Configurations close to the α-Lattice are the *quasi-α-Lattice* that introduces a tolerance in the definition of (3):

$$-\delta \le \|\boldsymbol{p}_i - \boldsymbol{p}_j\| - d \le \delta, \quad \forall (i, j) \in \mathcal{E}(\boldsymbol{p}), \tag{4}$$

where $\delta \in \mathbb{R}^+$ is a tolerance value. An example of such a structure is shown in Fig. 1b, where the dashed lines represent segments with length different from the lattice scale.

A measurement of the distance between the quasi-α-Lattice and the α-Lattice for a certain configuration $\boldsymbol{p}$ is given by an index defined as *deviation energy*:

$$E(\boldsymbol{p}) = \frac{1}{|\mathcal{E}(\boldsymbol{p})| + 1} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \left(\|\boldsymbol{p}_i - \boldsymbol{p}_j\| - d\right)^2. \tag{5}$$

It is worth noting that zero is the global minimum of such an index and is achieved for α-Lattice geometries.

In this paper, each node is modeled as a first-order dynamical (or single integrator) system. Thus, the equation of motion of the ith node is

$$\boldsymbol{v}_i = \boldsymbol{u}_i, \quad i \in \mathcal{V}, \tag{6}$$

where $\boldsymbol{v}_i \in \mathbb{R}^l$ is the velocity of each node, and $\boldsymbol{u}_i \in \mathbb{R}^l$ is its control input. The flocking problem consists of finding a control law $\boldsymbol{u}_i$ that drives the swarm to a quasi-α-Lattice structure with a desired level of tolerance δ.

3 The NSB control for multi-robot systems

In a general task, the accomplishment of several behaviors at the same time is required. For example, in a move-to-target movement in the presence of an obstacle, the robot has to achieve two behaviors: moving toward the target and keeping a safe distance from the obstacle. According to the common behavioral approaches, a high-level task commanded to the robotic system is usually decomposed into elementary *behaviors* that have to be simultaneously managed and arranged to elaborate the robots' motion directives. A possible technique to handle the behaviors' composition has been proposed by Bishop (2003) and Bishop and Stilwell (2001), which consists of assigning a relative priority to the single behaviors by resorting to the task-priority inverse kinematics introduced by Maciejewski (1988) and Nakamura et al. (1987) for ground-fixed redundant robotic manipulators. Nevertheless, as discussed by Chiaverini (1997), in the presence 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 idea is developed in Antonelli and Chiaverini (2003) in the framework of the singularity-robust task-priority inverse kinematics originally presented by Chiaverini (1997). This control approach, namely the Null-Space-based Behavioral control, has been then analyzed in the framework of behavior-based approaches for the control of a single autonomous vehicle in Antonelli et al. (2008a) and of multi-robot systems in Antonelli et al. (2008c). In this paper, the NSB approach is used in a decentralized architecture to individually control each robot. In detail, the NSB approach, implemented on each robot, is based on only local information, that is, the relative positions with respect to the neighbors and to the rendez-vous point. However, in practical impementations, it might be necessary to acquire the absolute position of the robot to estimate its relative position with respect to the rendez-vous point.

The task of the robot is decomposed into elementary behaviors, and, for each of them, a suitable function is defined. By defining as $\boldsymbol{\sigma} \in \mathbb{R}^m$ the generic variable to be controlled by the ith robot (m is the generic task dimension), it results in

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}_i, \boldsymbol{p}_i - \boldsymbol{p}_j), \tag{7}$$

where $\boldsymbol{p}_i \in \mathbb{R}^l$ is the position of the ith robot, and $\boldsymbol{p}_i - \boldsymbol{p}_j$ (where $\boldsymbol{p}_j \in \mathcal{N}_i$) is the relative displacement of the ith robot with respect to its generic neighboring robot.

Considering the neighboring robots as static, the corresponding differential relationship is

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{f}(\boldsymbol{p}_i, \boldsymbol{p}_i - \boldsymbol{p}_j)}{\partial \boldsymbol{p}_i} \boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{p}_i)\boldsymbol{v}_i, \tag{8}$$

where $\boldsymbol{J} \in \mathbb{R}^{m\times l}$ is the configuration-dependent behavior Jacobian matrix, and $\boldsymbol{v}_i \in \mathbb{R}^l$ is the robot velocity. An effective way to generate the control input for the robot starting from desired values $\boldsymbol{\sigma}_d(t)$ of the behavior function is to act at the differential level by inverting the (locally linear) mapping of (8); in fact, this problem has been widely studied in robotics (see, e.g., Siciliano 1990). A typical requirement is to pursue minimum-norm velocity, leading to the least-squares solution

$$\boldsymbol{u}_i = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}}_d, \tag{9}$$

where $\boldsymbol{J}^{\dagger}$ is the pseudo-inverse of the behavior Jacobian matrix $\boldsymbol{J}$ (for a low-rectangular full-rank Jacobian matrix, the pseudo-inverse is given by $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1}$).

However, discrete-time integration of the robot's reference velocity, needed for on-line implementation on digital devices, would result in a numerical drift of the reconstructed robot's position; the drift can be counteracted by a so-called Closed-Loop Inverse Kinematics

(CLIK) version of the algorithm, namely,

$$\boldsymbol{u}_i = \boldsymbol{J}^{\dagger}\big(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda}\widetilde{\boldsymbol{\sigma}}\big), \tag{10}$$

where $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains, and $\widetilde{\boldsymbol{\sigma}}$ is the behavior error defined as $\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$. Thus, the Null-Space-based Behavioral control intrinsically requires a differentiable analytic expression of the behaviors defined, so that it is possible to compute the required Jacobian matrices.

In the general case, the task for the single ith robot is composed of multiple behaviors; therefore, its motion reference is composed by merging the motion reference obtained by considering the single behaviors as acting alone. In particular, each behavior motion reference is designed so as to achieve its specific goal. However, it is generally impossible that a single motion command to the robot can accomplish all the goals at the same time. Therefore, when a motion command cannot simultaneously reduce the values of all the behavior functions, there is a *conflict* among the behaviors that must be solved by a suitable policy.

With respect to the main behavior-based approaches, the NSB approach presents a new *behavioral coordination* technique, that is, the way the single behavior outputs are composed to build the motion command for the robots. From a general point of view, the different solutions to the problem of behavioral coordination, as discussed in Arkin (1998), can be basically cast either in the frame of *competitive methods* or in the frame of *cooperative methods*. In the competitive methods, the behaviors are running in parallel in a distributed architecture, and at each time instant only one behavior is active by suppressing the other behaviors depending on the relative priorities; obviously, the robotic system is moving under the guidance of the sole active behavior. In the cooperative methods, by contrast, there is the need for a supervisor that elaborates each elementary behavior as if it were alone and builds the overall solution as the weighted sum of all the motion commands resulting from the single elementary behaviors; in addition, on the basis of sensory information, the supervisor can dynamically change the relative importance of the behaviors by changing the vector of weight gains. The layered architecture, proposed in Brooks (1986), is a classical example of a competitive method while the motor schema control (Arkin 1989) is one of the cooperative methods for behavioral approaches. The Null-Space-based Behavioral control, instead, uses a priority-based logic to combine multiple behaviors; in particular, it uses the null-space projectors to delete the output components of the lower priority behaviors that conflict with the higher-priority ones. Concerning the flocking problem, the NSB control approach differs from the other control approaches proposed in the literature in the way the single elementary behaviors are managed and combined. For instance, in the paper by Olfati-Saber (2006), the single elementary behaviors are combined following a potential approach and, from a behavior-based control point of view, implemented as a sort of cooperative control strategy; that is, the outputs of the single behavior functions are combined as a weighted sum to elaborate the final motion reference for the robot.

Using (10), the single behavior motion command is a velocity computed as

$$\boldsymbol{u}_{i,k} = \boldsymbol{J}_k^{\dagger}\big(\dot{\boldsymbol{\sigma}}_{k,d} + \boldsymbol{\Lambda}_k\widetilde{\boldsymbol{\sigma}}_k\big), \tag{11}$$

where k denotes the kth behavior. Let us further define as

$$\boldsymbol{N}_k = \big(\boldsymbol{I} - \boldsymbol{J}_k^{\dagger}\boldsymbol{J}_k\big)$$

the null space projector of the kth behavior. If the subscript k denotes the degree of priority of the behavior with, for example, behavior 1 being the highest-priority one, in the case of

3 behaviors (Chiaverini 1997; Mansard and Chaumette 2007), (10) becomes

$$\boldsymbol{u}_i = \boldsymbol{u}_{i,1} + \boldsymbol{N}_1 \boldsymbol{u}_{i,2} + \boldsymbol{N}_{12} \boldsymbol{u}_{i,3}, \tag{12}$$

where $\boldsymbol{N}_{12}$ is the null-space projector obtained by stacking the Jacobians corresponding to the behaviors 1 and 2. Iterating this procedure, it is possible to extend the technique to the desired number of behaviors.

In this way, the Null-Space-based Behavioral control always executes the highest-priority behavior. The lower-priority behaviors, on the other hand, are executed 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 only one single behavior can be executed at any given time, and to the cooperative approaches, where the use of a linear combination of each single behavior's output has as a result that no single behavior can be exactly executed. A deeper theoretical and experimental comparison among these approaches is performed in Antonelli et al. (2008a).

It is worth to mention that the computational complexity of this approach is mainly related to the DOFs of the robotic system. The NSB approach requires a matrix inversion for the null space projection. The decentralized implementation on mobile robots is thus computationally inexpensive.

4 Flocking via the NSB approach

In this paper, the flocking problem described in Sect. 2 is solved via the NSB approach by defining several local behavior functions and by implementing the control strategy in each robot. Each robot has its own local supervisor that is in charge of dynamically selecting the active behaviors and deciding their priority orders to properly perform the individual task.

In the following, the definitions of the behavior functions and the details of the supervisor are presented.

4.1 Behavior definitions

Two behaviors are sufficient to generate flocking behavior in a group of robots in the presence of a rendez-vous point. An additional behavior is required in the case of the presence of obstacles. Behaviors that rely only on local information are defined for each robot.

Lattice formation behavior This behavior function $\sigma_l \in \mathbb{R}$ is aimed at keeping the generic ith robot at a constant distance (the lattice scale) from its neighbors $\boldsymbol{p}_j \in \mathcal{N}_i$:

$$\sigma_l = \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \quad \text{with } \sigma_{l,d} = d. \tag{13}$$

Its Jacobian $\boldsymbol{J}_l \in \mathbb{R}^{1\times 3}$ and Null-space projector $\boldsymbol{N}_l \in \mathbb{R}^{3\times 3}$ are defined as

$$\begin{aligned} \boldsymbol{J}_l &= \hat{\boldsymbol{p}}_{ij}^{\mathrm{T}}, \\ \boldsymbol{N}_l &= \boldsymbol{I} - \hat{\boldsymbol{p}}_{ij} \hat{\boldsymbol{p}}_{ij}^{\mathrm{T}}, \end{aligned}$$

where

$$\hat{\boldsymbol{p}}_{ij} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}.$$

Moving to rendez-vous behavior The behavior function $\boldsymbol{\sigma}_r \in \mathbb{R}^3$ is aimed at making the robots converge to the same rendez-vous point. As it will be shown in the next sessions, the use of a common rendez-vous point makes it possible to create connected graphs. In fact, all the small flocks that are generated by the initial positioning of the robots move toward the rendez-vous point to create a unique aggregate flock. The definition of the behavior function is simply given by the robot position

$$\boldsymbol{\sigma}_r = \boldsymbol{p}_i \quad \text{with } \boldsymbol{\sigma}_{r,d} = \boldsymbol{p}_{\text{rv}}, \tag{14}$$

where $\boldsymbol{p}_{\text{rv}} \in \mathbb{R}^3$ is the rendez-vous point. The 3×3 Jacobian is simply the identity matrix, and the null space projector is a 3×3 null matrix.

Obstacle avoidance behavior Obstacle avoidance for autonomous robots is a mandatory task and, resorting to the NSB approach, it has been deeply discussed in previous papers such as, e.g., Antonelli and Chiaverini (2006) and Antonelli et al. (2008a). Not surprisingly, the obstacle avoidance behavior function is formally equal to the lattice formation behavior:

$$\sigma_o = \|\boldsymbol{p}_i - \boldsymbol{p}_o\|, \quad \text{with } \sigma_{o,d} = d_o, \tag{15}$$

and

$$\begin{aligned} \boldsymbol{J}_o &= \hat{\boldsymbol{p}}_{io}^{\mathrm{T}} \\ \boldsymbol{N}_o &= \boldsymbol{I} - \hat{\boldsymbol{p}}_{io}\hat{\boldsymbol{p}}_{io}^{\mathrm{T}} \end{aligned} \quad \text{with } \hat{\boldsymbol{p}}_{io} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_o}{\|\boldsymbol{p}_i - \boldsymbol{p}_o\|},$$

where $\boldsymbol{p}_o$ is the position of the obstacle, and d_o the desired distance from it.

It is worth noting that the previously defined functions represent elementary behaviors for each robot. In this sense, a behavior function can be used several times if needed. As an example, a robot can implement the lattice behavior with respect to several different neighboring robots in its sensing range.

4.2 Supervisor

The supervisor is a higher-level function that is in charge of selecting the active behaviors and their priorities. The supervisor decides which behavior has to be activated depending on the environmental condition (e.g., presence of obstacles in the sensory range) and on the system status (e.g., relative position with respect to neighboring robots). Moreover, the supervisor limits the number of active behaviors considering the dimensions of the behavior functions to avoid requiring the fulfillment of an overall task with dimension larger than that of the available Degrees of Freedom (DOFs) of the system. That is, when the higher-priority behaviors span all the DOFs of the dynamic system, it results useless adding further behaviors since the global null-space of the higher-priority behaviors is an empty space and the lower-priority behaviors would not take effect. Further consideration on choice of the maximum number of active behaviors depending on the DOFs of the system can be found in Antonelli et al. (2008a).

Each of the robots is only aware of the robots within its sensing range. To decide the active behaviors, each ith 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 three-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 $\mathcal{N}_i$).
- Lattice behavior with respect to the robot $\boldsymbol{k}_i(2)$ (if there are two robots in $\mathcal{N}_i$).
- Lattice behavior with respect to the robot $\boldsymbol{k}_i(3)$ (if there are three or more robots in $\mathcal{N}_i$).

Table 1 Behaviors activated by the ith robot supervisor in the presence of a unique neighbor. The rendez-vous behavior dimension is 2 in the 2D case, 3 in the 3D case

Priority	Behavior	Dimension
1	Lattice behavior with respect to the jth robot	1
2	Rendez-vous behavior	2/3

Table 2 Behaviors activated by the ith robot supervisor in the presence of multiple neighbors and an obstacle

Priority	Behavior	Dimension
1	Obstacle avoidance	1
2	Lattice behavior with respect to the robot $\boldsymbol{k}_i(1)$	1
3	Lattice behavior with respect to the robot $\boldsymbol{k}_i(2)$	1

– Moving to rendez-vous behavior (if required in the task).
– Obstacle avoidance behavior (if $\|\boldsymbol{p}_i - \boldsymbol{p}_o\| < r$).

These behaviors need to be properly arranged into a priority order. A trivial situation arises when flocking is required without a rendez-vous point and the set $\mathcal{N}_i$ is empty, in which case the robot obviously stays still.

Let us first consider the case of the absence of obstacles in the sensor range. In this case, the supervisor computes the Lattice behaviors, assigning the highest priority to the closest robot. Since the Lattice behavior is one-dimensional ($\sigma_l \in \mathbb{R}$), if at least three robots belong to $\mathcal{N}_i$, the moving-to-rendez-vous behavior is discarded; otherwise, it is added as the lowest in priority. It is worth noting that, even if more than three robots belong to $\mathcal{N}_i$, for the approach presented here, it is sufficient to consider only the closest three and not all of them as for potential approaches.

Let us now consider one obstacle in the interaction range of the robot. The supervisor firstly computes the desired velocity disregarding the obstacle, then checks if the robot would collide with the obstacle or not. In the latter situation nothing is changed with respect to the non-obstacle case. If, on the other hand, there is the chance to collide with the obstacle, then the obstacle-avoidance behavior is selected as the highest-priority behavior, and all the other behaviors are correspondingly lowered in priority. The last behavior is eventually removed if the sum of the behaviors' dimension is larger than three.

For the sake of clarity, let us imagine a situation where the ith robot has only one jth robot in its interaction range. Its supervisor would then consider only the behaviors shown in Table 1. On the other hand, if several robots and the obstacle are inside the interaction range, the supervisor would output the behaviors shown in Table 2.

It is worth mentioning that in Olfati-Saber (2006), as well as in this paper, each robot needs to know its relative position with respect to the other robots present in a set $\mathcal{N}_i$ that is simply a sphere around it. This is reasonable for robotic systems but not for a flock in nature mainly characterized by directional sensing such as, e.g., eye-based vision. Future research might consider anisotropic sets $\mathcal{N}_i$ and proper behaviors that take these into account.

5 Simulation results

Extensive simulations were run using Scilab (http://www.scilab.org/) to test the performance of the algorithm in different conditions; in particular, the tests concerned the absence/presence of a common (eventually moving) rendez-vous point, the presence/absence of obstacles, and the 2D/3D implementation. All the simulations were run with robots start-

Table 3 Parameters used in the simulation case studies

Number of robots	between 5 and 100
Sampling time	0.1 s
Lattice scale	$d = 7$ m
Safety distance from obstacle	$d_o = 7$ m
Interaction range	$r = 9.8$ m
Lattice ratio	$\kappa = \frac{r}{d} = 1.4$
Lattice formation behavior gain	$\lambda_l = 1.5$
Rendez-vous behavior gain	$\lambda_r = 0.5$
Obstacle avoidance behavior gain	$\lambda_o = 0.5$

ing from random positions. Results of several case studies are presented in the following sections, while the videos of the simulations can be downloaded from the laboratory URL (address in the affiliation) or available in the on-line supplementary material. Moreover, in Sect. 5.6, a quantitative analysis of the simulation results will be presented.

In all of the simulations, the parameters are defined in Table 3, where λ_l, λ_r, and λ_o are the NSB gains defined in (10) for the lattice formation, moving to rendez-vous point and obstacle avoidance behaviors respectively (referring to (10), $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}$, where λ is a scalar, and $\boldsymbol{I}$ is the identity matrix with the dimension of the behavior function).

5.1 Two-dimensional case without obstacles and rendez-vous point

In the first simulation, a team of 40 robots is utilized. The term has to move in a 2D environment free from obstacles and without a common rendez-vous point. Figure 2 shows four snapshots of the simulation; it can be observed that, due to the absence of a rendez-vous point, the robots do not have a common reference point to which they can converge, and their motion only depends on the relative position of neighbors. Thus, the robots cannot converge to a connected lattice configuration; instead, they form small independent groups (see the video 11721_2009_36_MOESM5_ESM.mpg in the supplementary material for a similar simulation). This kind of fragmentation of the reticular structure was observed also by Olfati-Saber (2006) with a different algorithm.

5.2 Two-dimensional case without obstacles and with a static rendez-vous point

In the second simulation, a team of 100 robots starting from a random configuration is required to move in the presence of a static rendez-vous point. Figure 3 shows the final configuration of the team after 20 s. It is worth noting that, in this case, an α-lattice structure is formed. Moreover, the robots assume a connected lattice structure without fragmentation. See the video 11721_2009_36_MOESM7_ESM.mpg in the supplementary material for a similar simulation.

5.3 Two-dimensional case with obstacles and a static rendez-vous point

In the case of flocking in the presence of a static rendez-vous point in an environment with obstacles, the robots tend to achieve an α-lattice structure, but, since obstacle avoidance has the highest priority, they cannot form a regular lattice structure in order to keep a safe distance from the obstacle. Figure 4 shows four snapshots of a simulation involving 30 robots with a stationary rendez-vous point and a single obstacle. See the video 11721_2009_36_MOESM6_ESM.mpg in the supplementary material for a similar simulation.

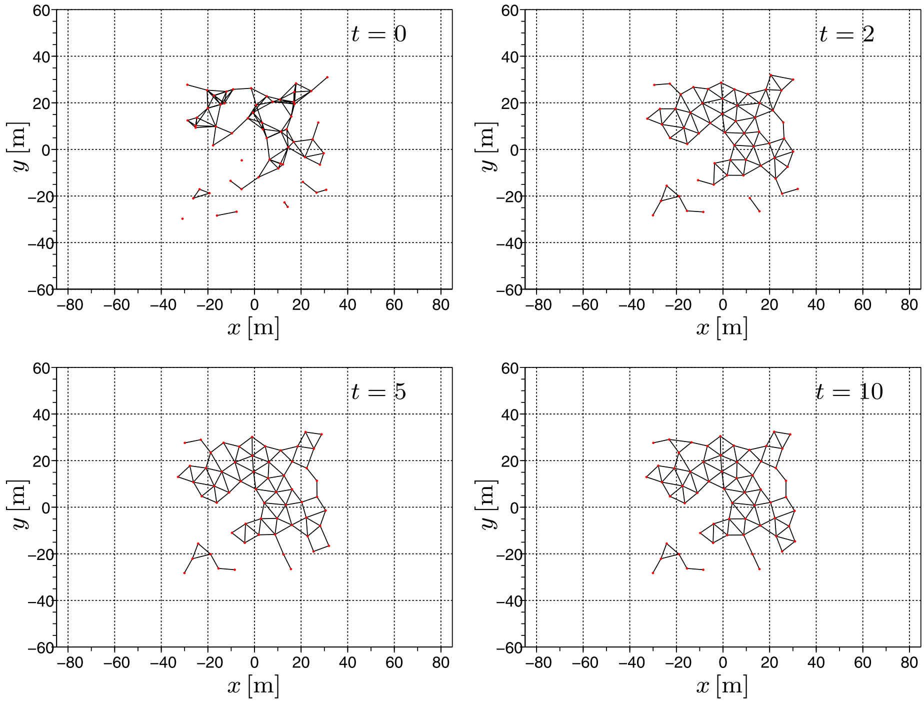


Fig. 2 Snapshots for 40 robots flocking in the absence of rendez-vous point and obstacles; as expected, the flocking is partial or fragmented. The *dots* represent the robots, and the *solid lines* the connections

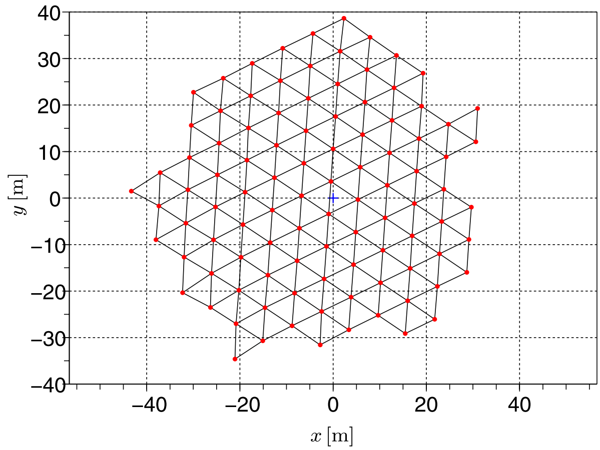


Fig. 3 Steady-state configuration for the 2D case study, with rendez-vous point, without obstacles and with 100 robots

5.4 Two-dimensional case without obstacles and with a moving rendez-vous point

In this simulations, a team of 40 robots is required to move in the presence of a rendez-vous point moving with a velocity of $\dot{\boldsymbol{p}}_{\mathrm{rv}} = [3\ 0]^{\mathrm{T}}$ m/s. Figure 5 shows four snapshots of the simulation. Flocking is achieved, and the robots follow the moving rendez-vous point. Notice that the group does not move as a rigid formation tracking the target but rather prop-

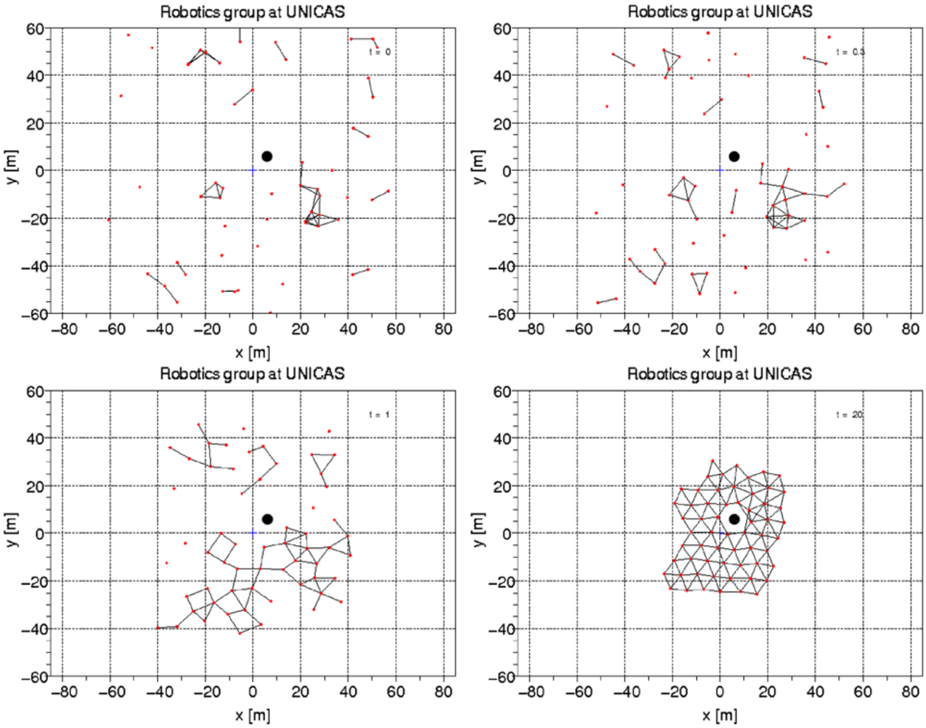


Fig. 4 Snapshots for 30 robots flocking in the presence of a static rendez-vous point and an obstacle. The *dots* represent the robots, the *solid lines* represent the connections, the *cross* at the origin represents the rendez-vous point, and the *biggest dot* represents the obstacle

agates its movement along the group. This result is due to the local nature of the single robot controllers and the selected priorities for the behaviors: the group moves to execute the lattice behavior with a higher priority than the moving-to-rendez-vous behavior. It can also be observed that the group exhibits a sort of *compression/expansion* in the direction of rendez-vous movement. See the video 11721_2009_36_MOESM4_ESM.mpg in the supplementary material for a similar simulation.

5.5 Three-dimensional case without obstacles and with static rendez-vous point

In this simulation, a team of 15 robots is required to move, in the presence of a static rendez-vous point, in the 3D space free from obstacles. Figure 6 shows four snapshots of the simulation. Flocking is successfully achieved in this case. Due to the difficulty in reading the snapshots, the reader is referred to the video of the simulation flock_3D.gif (available in the online supplementary material) in order to understand the motion of the robots in a better way.

5.6 Quantitative analysis of simulations

The results presented above are selected examples of an intensive simulative analysis aimed in testing the efficiency of the proposed approach under different conditions. In this section, we present a quantitative analysis of the simulations focusing on a specific case study of 2D

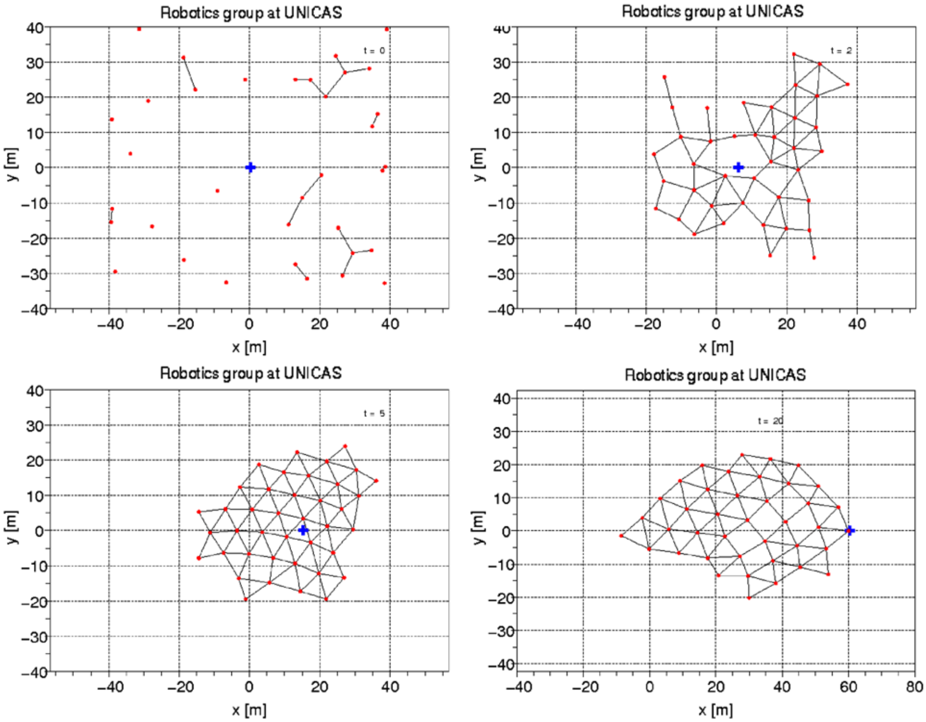


Fig. 5 Snapshots for 40 robots flocking in the presence of a moving rendez-vous point and without obstacles. The *dots* represent the robots, the *solid lines* the connections, and the *cross* initially at the origin the rendez-vous point

robots flocking in the presence of a static rendez-vous point in an environment free from obstacles; the mission parameters have been chosen as in Table 3.

The first simulation is performed to show the relationship between the time of convergence to form a lattice structure and the size of the swarm. Since the initial configuration is a critical parameter that influences the transient behavior of the team, 20 simulations with randomly placed robots were run with teams of fixed number of robots. A uniform probability density function was utilized with ranges related to the square root of the swarm size to choose the initial locations of the robots; this choice allowed us to impose the same density of robots in the area and thus to have homogeneous data to work with. To properly compare results from different simulations, it is also necessary to *detect* when the flock has reached a steady state. Several metrics have been proposed in the literature to give a mathematical measurement of how far a configuration is from an α*-Lattice* structure such as, for example, the deviation energy already shown in (5), the *social entropy* proposed by Balch (2000), and the *cohesion radius* proposed by Gu and Hu (2008). In our study, the deviation energy is used; in particular, the flocking is considered to reach a steady-state condition when the index is smaller than a given threshold.

Figure 7 shows a bar graph of the results. The x-axis shows the number of robots, while the y-axis shows the minimum, maximum, and average convergence times for the 20 simulations. The left (grey) columns represent the simulations without sensing noise, while the right columns show the simulations with a sensor noise in the relative position measurement having a uniform probability density function between ranges ± 20 cm (this value corre-

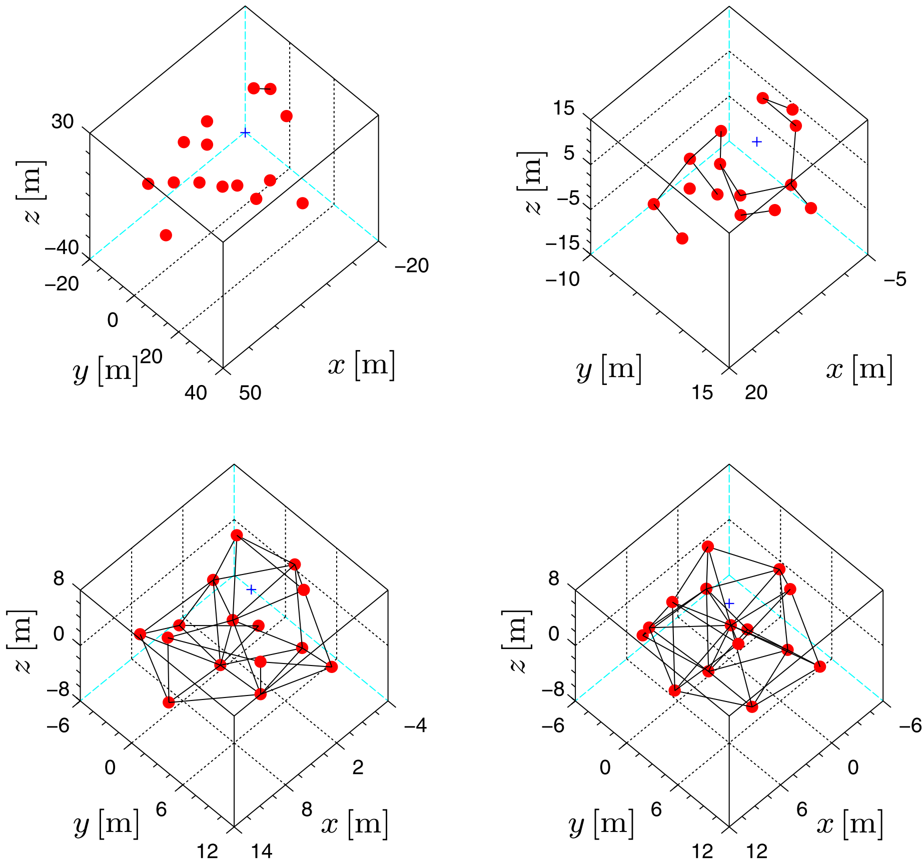
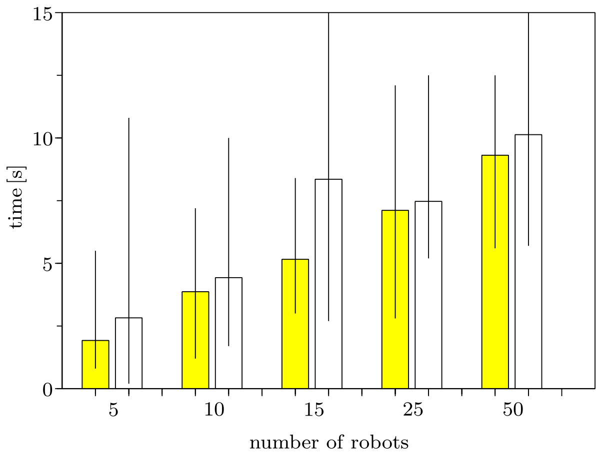


Fig. 6 Snapshots for 15 robots in 3D. The *dots* represent the robots, the *solid lines* the connections, and the *cross* at the origin the rendez-vous point

Fig. 7 Convergence time for teams composed of an increasing number of robots. For each fixed size of the team, the simulations have been repeated 20 times starting from different random configurations. For each simulation set, the minimum, average, and maximum values of the convergence time are reported. The *left (grey) columns* represent the convergence times of simulations without sensing noise, while the *right columns* shows the results of simulations with sensor noise in the relative position measurement



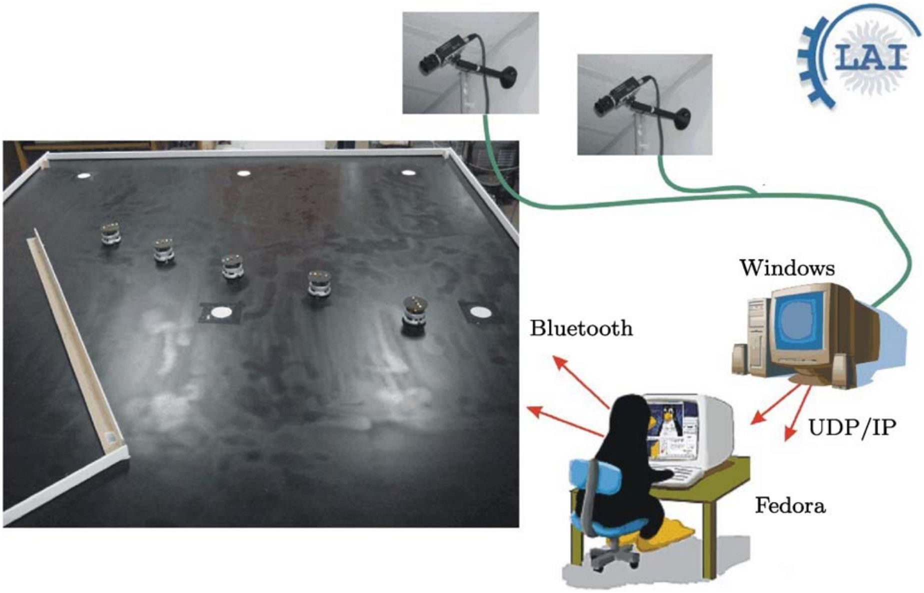


Fig. 8 Sketch of the multi-robot set-up available at LAI (Laboratorio di Automazione Industriale) of the Università degli Studi di Cassino

sponds to 2% of the scale length). It can be observed that the size of the group influences the transient behavior of flocking. When the robots are in a quasi-*alpha*-lattice configuration, a group of locally interacting robots is formed, and the movement of one robot propagates through all the remaining ones; thus, there is a long period of small adjustments among the robots before *alpha*-lattice configuration is reached. It can be observed that the presence of noise has an adverse effect on the average convergence time of flocking.

6 Experiments with real robots

In this section, the experimental set-up and the results of flocking with real robots are presented.

6.1 Experimental set-up

We use seven Khepera II mobile robots (manufactured by K-team) available in the LAI (Laboratorio di Automazione Industriale) of the Università degli Studi di Cassino. These are differential-drive mobile robots, with a unicycle-like kinematics and an approximate diameter of 8 cm. Each robot can communicate with a remote PC using the Bluetooth communication protocol.

The Khepera II robots do not have the capability to self-localize and estimate the relative positions of the neighboring robots with sufficient accuracy. In order to focus the experiments on the validation of the control algorithm, an external position measuring system was used: a vision-based system using two CCD cameras, a Matrox Meteor-II frame grabber (manufactured by Matrox Electronic Systems Ltd) and a custom C++ image-processing software. The acquired images are 1024×768 RGB bitmaps. The measurement error has

Table 4 Parameters used in the experiments with real robots

	Fixed rendez-vous	Moving rendez-vous
Number of robots	7	7
Lattice scale	$d = 25\,\text{cm}$	$d = 25\,\text{cm}$
Interaction range	$r = 30\,\text{cm}$	$r = 40\,\text{cm}$
Lattice ratio	$\kappa = \frac{r}{d} = 1.2$	$\kappa = \frac{r}{d} = 1.6$
Lattice formation behavior gain	$\lambda_l = 0.3$	$\lambda_l = 0.3$
Rendez-vous behavior gain	$\lambda_r = 0.5$	$\lambda_r = 0.2$

an upper bound of ∼0.5 cm and ∼1 deg. A remote PC receives from the vision system the position measurements at a sampling time of 80 ms and implements the NSB control for each of the robots, that is, several independent controllers are implemented on the remote PC; each of the controllers has access only to the corresponding robot's position and to its neighbors' relative positions. In this way the decentralized controllers are implemented on the central unit by *filtering* the inaccessible information for each robot. Once the NSB controller outputs the desired linear velocities for each robot, a heading controller is implemented (Oriolo et al. 2002) to obtain the wheels' desired velocities to steer the robot in the desired direction and move with the desired forward velocity. The remote PC sends the desired velocities of the wheels to each robot through Bluetooth with a sampling time of $T = 80$ ms. The controller of the wheels velocity is a PID developed by the manufacturer. Saturations of 40 cm/s and 100 deg/s have been introduced for the linear and angular velocities, respectively. Moreover, the encoder resolution is such that a quantization of ∼0.8 cm/s and ∼9 deg/s are experienced.

6.2 Experimental results

Two different kinds of flocking experiments were performed: one with fixed rendez-vous point and the other with moving rendez-vous point (both without obstacles). In both cases, the initial robot configuration was random. The parameters used are shown in Table 4.

The parameters were empirically chosen based on the system dynamics and on the simulation results, while the fine-tuning was performed by trial and error.

Figures 9a and 9c show two snapshots corresponding to the initial and final configuration of an experiment run with 7 robots and a fixed rendez-vous point. The respective graphical elaborations (Figs. 9b and 9d) better illustrate the robots' connections; in particular, the cone shows the rendez-vous point, while the lines show the neighborhood relations among the robots. Several snapshots of the experiment run with 7 robots and a moving rendez-vous point are shown in Fig. 10. Despite the presence of nonholonomicity, dynamics, communication delays, and so on, it can be observed that the flocking behavior is successfully achieved in both scenarios. The videos of both the experiments are available on-line as supplementary material (videos 11721_2009_36_MOESM1_ESM.mpg and 11721_2009_36_MOESM2_ESM.mpg). The hand-camera videos, together with the graphical elaboration, allow a better understanding of the team dynamic behavior during the experiments.

As for most of the control approaches for swarm robotics, a metric that allows us to properly evaluate the swarm controller performances is lacking. Even though the effectiveness of the approach can be appreciated from experimental evidence of the flocking behavior, no quantitative measurement expressing how good the behavior is can be given. The identification of an effective metric will be object of future research; however, to appreciate the

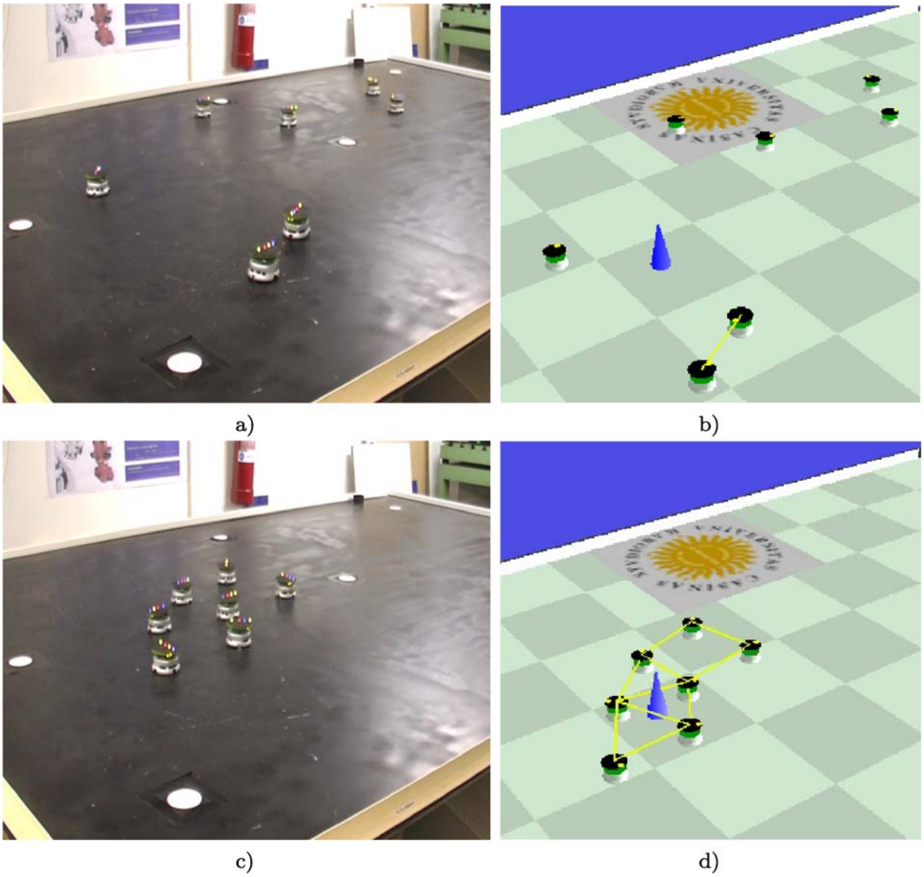


Fig. 9 Snapshots of the initial and final configurations (respectively, (**a**) and (**c**)) and their graphical elaborations (respectively, (**b**) and (**d**)) of an experiment where seven robots flock around a fixed rendez-vous point. In the left figures, the *white* discs are the markers used for the calibration of the vision system; in the right figures, the *cone* represents the position of the rendez-vous point

proposed technique, it is worth noting that the NSB approach does not require communication among robots or neighbor velocity and orientation estimation. Thus, the proposed approach can be used to control robots equipped with common sensors for mobile robots without the need of communication devices.

7 Conclusions

In recent years, the Null-Space-based Behavioral control approach has been applied to a wide range of robotic systems; its main advantage is the possibility to take the advantages of behavioral approaches in terms of flexibility and possibility to manage dynamic tasks, together with a rigorous analytical approach that allows one to extrapolate mathematical convergence properties. On the other hand, the NSB approach always requires that the behavior can be described by means of an analytical function so that a Jacobian can be derived. In the case of multi-robot systems, problems such as formation control, escorting a target,

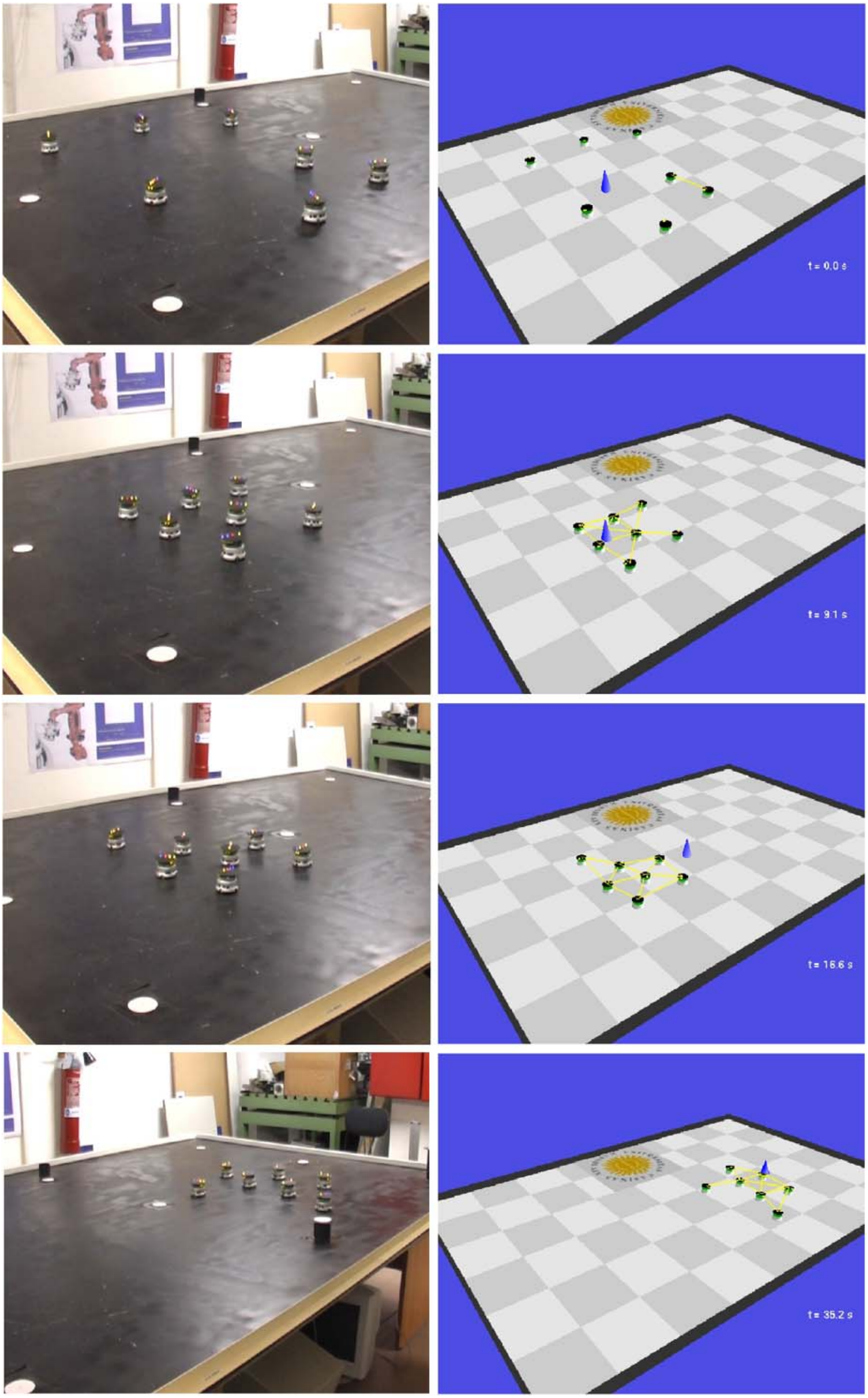


Fig. 10 Several snapshots of a flocking experiment with moving rendez-vous point. In the right figures, the cone represents the position of the rendez-vous point

and reconfiguration as a mobile ad-hoc network have been successfully achieved. In this paper, the flocking behavior of a group of robots has been addressed using the Null-Space-based Behavioral control. Flocking in the presence of a common rendez-vous point and/or in the presence of obstacles has been discussed and verified by 2D/3D-numerical simulations and 2D experiments with real robots. It has been shown that very simple behavior functions activated by a supervisor can make the overall system successfully execute the flocking behavior. Future research will focus on the execution and testing of the flocking behavior implemented in a fully decentralized set-up, thus overcoming the limitations of our current experimental set-up. Moreover, further investigations will concern the use of anisotropic sensors to recognize and localize the neighboring robots, like vision systems with limited view cones.

References

Antonelli, G., Arrichiello, F., Chakraborti, S., & Chiaverini, S. (2007). Experiences of formation control of multi-robot systems with the Null-Space-based Behavioral control. In *Proceedings 2007 IEEE international conference on robotics and automation* (pp. 1068–1073). New York: IEEE Press.

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2008a). The null-space-based behavioral control for autonomous robotic systems. *Journal of Intelligent Service Robotics*, *1*(1), 27–39.

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2008b). Stability analysis for the null-space-based behavioral control for multi-robot systems. In *Proceedings 47th IEEE conference on decision and control and 8th European control conference* (pp. 2463–2468). New York: IEEE Press.

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2008c). 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.

Antonelli, G., & Chiaverini, S. (2003). Kinematic control of a platoon of autonomous vehicles. In *Proceedings 2003 IEEE international conference on robotics and automation* (pp. 1464–1469). New York: IEEE Press.

Antonelli, G., & Chiaverini, S. (2006). Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, *22*(6), 1285–1292.

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

Arkin, R. C. (1998). *Behavior-based robotics*. Cambridge: MIT Press.

Balch, T. (2000). Hierarchic social entropy: an information theoretic, measure of robot group, diversity. *Autonomous Robots*, *8*(3), 209–238.

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

Bishop, B. E., & Stilwell, D. J. (2001). On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles. In *Proceedings 2001 IEEE international conference on control applications* (pp. 823–828). New York: IEEE Press.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14–23.

Cao, Y., Fukunaga, A. S., & Kanhg, A. B. (1997). Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, *4*(1), 7–27.

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

Cortes, J., Martinez, S., & Bullo, F. (2006). Robust rendez-vous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, *51*(8), 1289–1298.

Gu, D., & Hu, H. (2008). Using fuzzy logic to design separation function in flocking algorithms. *IEEE Transactions on Fuzzy Systems*, *16*(4), 826–838.

Hsieh, M. A., Kumar, V., & Chaimowicz, L. (2008). Decentralized controllers for shape generation with robotic swarms. *Robotica*, *26*(05), 691–701.

Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, *48*(6), 988–1001.

Ji, M., & Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, *23*(4), 693–703.

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

Mansard, N., & Chaumette, F. (2007). Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics and Automation*, *23*(1), 60–72.

Martinez, S., Cortes, J., & Bullo, F. (2007). Motion coordination with distributed information. *IEEE Control Systems Magazine*, *27*(4), 75–88.

Matarić, M. J. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, *16*(2), 321–331.

Moshtagh, N., & Jadbabaie, A. (2007). Distributed geodesic control laws for flocking of nonholonomic agents. *IEEE Transactions on Automatic Control*, *52*(4), 681–686.

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

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, *51*(3), 401–420.

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.

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

Ren, W., & Beard, R. W. (2008). *Communications and control engineering*. *Distributed consensus in multi-vehicle cooperative control*. Berlin: Springer.

Ren, W., Beard, R. W., & Atkins, E. M. (2007). Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, *27*(2), 71–82.

Reynolds, C. (1987). Flocks, herd and schools: a distributed behavioral model. *Computer Graphics*, *21*(4), 25–34.

Siciliano, B. (1990). Kinematic control of redundant robot manipulators: a tutorial. *Journal of Intelligent Robotic Systems*, *3*(3), 201–212.

Tanner, H., Jadbabaie, A., & Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, *52*(5), 863–868.