Swarm of Robots Flocking via the Null-Space-based Behavioral Control

Gianluca Antonelli Filippo Arrichiello Stefano Chiaverini

*Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale*

Università degli Studi di Cassino

via G. Di Biasio 43, 03043 Cassino (FR), ITALY

{antonelli, f.arrichiello, chiaverini}@unicas.it

<http://webuser.unicas.it/lai/robotica/>

Abstract—In this paper the problem of flocking together multiple robots is considered. The capability of grouping the robots is achieved resorting to a behavior-based approach, namely the Null-Space-based Behavioral (NSB) control. In this framework, the assigned mission is accomplished by defining very simple behaviors for each robot and properly arranging them in priority. In particular, flocking is pursued in a decentralized manner, i.e., the elementary behaviors of each robot are defined so they only need local information concerning robot's neighbors. To test the performance of the method, the flocking behavior is analyzed in different cases, namely, in presence of a static/moving rendez-vous point, changing the size of the team and in presence/absence of measurement noise. Extensive simulations prove the effectiveness of the proposed technique.¹

I. INTRODUCTION

Many living beings exhibit collective behaviors in nature by implementing local control strategies. Among the different collective behaviors, flocking has fascinated researchers from several disciplines, e.g., physicists, social scientists, roboticists, animal psychologists, etc. By the term flocking it is meant the capability of many individuals to congregate and stay grouped as a result of individual actions. As discussed in [20], the flocking problem is also 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, flocking can be considered in Robotics a specific case of coordinated control of multiple robots that involves distribute sensing, exploration, coverage, search and rescue, etc; see [9] for a good overview.

In 1987, Reynolds published a seminal work in which a computer model for motion coordination of animals as bird flocks or fish schools was presented [25]. An extensive literature now exists that reports interesting results concerning the flocking problem. In [21], different solutions are investigated and their stability is analyzed; 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* for each robot. The work [12] 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 [19] surveys recent developments in modeling, analysis and design of distributed motion coordination algorithms for multi-robot systems. The work [17] presents a potential field control approach for swarm of robots collecting objects.

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, reaching an agreement regarding a certain quantity of interest that depends on the state of all the agents [22]. An overview of the information consensus is given in [24], [23], while the work [22] investigates consensus algorithms with emphasis on robustness, time-delays and performance guarantee. The work [15] shows how a consensus variable can be used to make the flock perform particular behaviors like formation keeping or rendez-vous. The work [14] develops the stability analysis of several decentralized strategies that achieve an emergent behavior.

Autonomous robotics has been strongly influenced by the paradigm of behavior-based control, introduced in the works [8], [7]. Using sensors to obtain information about the environment, behavior-based control approaches give the system the autonomy to operate in complex environments. Thus, they result useful and successful for the control of both single robots and multi-robot systems, e.g., 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 Null-Space-based Behavioral (NSB) control presented in [3]. This approach, strongly related to the kinematic control presented in [5], [6], uses a hierarchy-based strategy to compose the elementary behaviors that concur to accomplish the mission of the multi-robot systems. In particular, in case of conflicting behaviors, the NSB filters out the lower priority behaviors' components that would conflict with the higher priority behaviors. The NSB approach takes the advantages

¹The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n. 231378.

of behavior-based approaches in the ease of design and the reactivity to unknown or changing conditions, while it presents a rigorous mathematical formulation that allows to extrapolate some analytical convergence properties [4]. Following the results in [2], [1], this paper presents the use of the NSB approach as a decentralized control technique to achieve the flocking mission in different conditions. Each robot requires only local information such as the relative position to its neighbors and, only in case of a rendez-vous, its global position. Extensive simulations, assuming two-dimensional point-mass robots, prove the effectiveness of the proposed approach.

II. THE FLOCKING PROBLEM

In this paper, flocking of a swarm of robots is considered as the mission of grouping them according to a *lattice* configuration as an emerging behavior obtained by implementing individual controllers on board of each robot. The robots can only sense their relative positions to their neighbors; moreover, when the team has to converge to a rendez-vous point, each robot also needs to know its absolute position. It is worth noticing that in the literature the same flocking problem is also approached under less strict assumptions, i.e., by sensing the neighbors' relative positions/velocities and/or by communicating with them. In the following, the basic notions on graph theory are briefly recalled to provide a tool to formalize the problem.

A *graph* G is a pair $(\mathcal{V}, \mathcal{E})$ that consists of a set of *vertices* $\mathcal{V} = \{1, 2, \dots, n\}$ and *edges* $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$. In this paper, an undirected graph will be considered, i.e., $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$. The scalar quantity $|\mathcal{V}|$ is the *order* and $|\mathcal{E}|$ is the *size* of the graph. The *adjacent matrix* $\mathbf{A}$ collects the information concerning the edges such that $a_{i,j} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E}$; for an undirected graph it is $\mathbf{A} = \mathbf{A}^T$. The set of neighbors of node i is defined as

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

In our case, the graph is related to the physical displacement of nodes in the Euclidean space. Thus, let us denote as $\mathbf{p}_i \in \mathbb{R}^l$ the position of each node, where $l=2$ in case of a material point moving on a surface, or $l=3$ in the case of a material point moving in the 3D-space. Then, the *configuration* of all the nodes of the graph is represented by the vector $\mathbf{p} \in \mathbb{R}^{ln}$ defined as

$$\mathbf{p} = [\mathbf{p}_1^T \quad \mathbf{p}_2^T \quad \dots \quad \mathbf{p}_n^T]^T.$$

A *framework*, or *structure*, is a pair $(G, \mathbf{p})$ that consists in a graph and the configuration of its nodes. To represent the relation between pairs of adjacent nodes in the configuration, the graph presents an edge between two nodes $(a_{i,j} \neq 0)$ if their Euclidean distance is lower than a threshold value r , called the *interaction range*. Thus, the neighbors relation of equation (1) is equivalent to:

$$\mathcal{N}_i = \{j \in \mathcal{V} : \|\mathbf{p}_i - \mathbf{p}_j\| < r\} \quad (2)$$

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

In order to define the desired flocking configuration, an α -*Lattice* structure is introduced as a framework that satisfies

$$\|\mathbf{p}_i - \mathbf{p}_j\| = d \quad \forall j \in \mathcal{N}_i, \quad (3)$$

where d is the lattice's *scale*. In words, an α -Lattice is the structure of a configuration characterized by the fact that all the edges of the graph have the same length (see Figure 1.a). Moreover, let us define the lattice's *ratio* as the value $\kappa = r/d$.

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

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

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

where $\delta \in \mathbb{R}$. An example of such a structure is shown in Figure 1.b, where the dashed lines have length different from the lattice's scale within the given tolerance.

A measurement of deviation of a quasi- α -Lattice from the corresponding α -Lattice for a certain configuration $\mathbf{p}$ is given by the index

$$E(\mathbf{p}) = \frac{1}{|\mathcal{E}(\mathbf{p})| + 1} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} (\|\mathbf{p}_i - \mathbf{p}_j\| - d)^2, \quad (5)$$

that is called *deviation energy*. Remarkably, this index has a global minimum at zero which is attained by the α -Lattice structure.

In this paper, the nodes are modeled as first-order systems (or single integrators), thus, their equation of motion is:

$$\mathbf{v}_i = \mathbf{u}_i \quad \forall i \in \mathcal{V},$$

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

III. NSB CONTROL FOR MULTI-ROBOT SYSTEMS

According to all behavior-based approaches, the overall mission for the team of autonomous robots is decomposed in elementary sub-problems (or *tasks*) that have to be simultaneously managed. For each task a suitable task function is defined and a motion control directive is elaborated. Then, the single tasks that compose the overall mission are organized in a proper priority order. Finally, the global motion control directive to the robots is elaborated by suitably combining the motion directives of the single tasks.

In detail, by defining as $\sigma \in \mathbb{R}^m$ the generic task variable to be controlled by the i -robot (m is the task dimension), it is:

$$\sigma = \mathbf{f}(\mathbf{p}_i, \mathbf{p}_i - \mathbf{p}_j), \quad (6)$$

where $\mathbf{p}_i \in \mathbb{R}^l$ is the i -robot position and $\mathbf{p}_i - \mathbf{p}_j$ (with $\mathbf{p}_j \in \mathcal{N}_i$) is the relative displacement of the i -robot from its neighboring j -robot.

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

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \mathbf{f}(\mathbf{p}_i, \mathbf{p}_i - \mathbf{p}_j)}{\partial \mathbf{p}_i} \mathbf{v}_i = \mathbf{J}(\mathbf{p}_i) \mathbf{v}_i, \quad (7)$$

where $\mathbf{J} \in \mathbb{R}^{m \times l}$ is the configuration-dependent task Jacobian matrix and $\mathbf{v}_i \in \mathbb{R}^l$ is the robot velocity. An effective way to generate motion references for the robot starting from desired values $\boldsymbol{\sigma}_d(t)$ of the task function is to act at the differential level by inverting the (locally linear) mapping of equation (7); in fact, this problem has been widely studied in robotics (see, e.g., [11] for a tutorial). A typical requirement is to pursue minimum-norm velocity, leading to the least-squares solution:

$$\mathbf{u}_i = \mathbf{J}^\dagger (\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}), \quad (8)$$

where $\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}$ (when $\mathbf{J}(\mathbf{q})$ is full rank), $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains and $\tilde{\boldsymbol{\sigma}}$ is the task error defined as $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$. It is worth noticing that the term $\boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}$ is added to counteract the numerical drift due to discrete-time integration.

When the mission is composed by multiple tasks the overall vehicle velocity is elaborated by properly merging the outputs of the single tasks. In particular, each task velocity is computed as if it were acting alone; then, before adding its contribution to the overall vehicle velocity, a lower-priority task is projected onto the null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it. Thus, on the analogy of eq. (8), the single task velocity is computed as

$$\mathbf{u}_{i,k} = \mathbf{J}_k^\dagger (\dot{\boldsymbol{\sigma}}_{k,d} + \boldsymbol{\Lambda}_k \tilde{\boldsymbol{\sigma}}_k), \quad (9)$$

where the subscript k denotes the generic k -th task. Let us further define as

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

the null space projector of the task k . If the subscript k also denotes the degree of priority of the task with, e.g., Task 1 being the highest-priority one, in the case of 3 tasks and according to [10], [18], then, the CLIK solution (8) is modified into

$$\mathbf{u}_i = \mathbf{u}_{i,1} + \mathbf{N}_1 \mathbf{u}_{i,2} + \mathbf{N}_{1,2} \mathbf{u}_{i,3} \quad (10)$$

where $\mathbf{N}_{1,2}$ is the null space projector obtained by stacking the Jacobians corresponding to the tasks 1 and 2.

Concerning the flocking problem, the NSB control approach differs from the other approaches proposed in the literature in the way the single elementary tasks are managed and combined. For instance, in the paper [21], the single elementary tasks 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 task functions are combined as a weighted sum to elaborate the final motion reference for the robot.

IV. FLOCKING VIA THE NSB APPROACH

The flocking problem described in Section II is solved here by defining several local task functions and by applying the NSB control strategy on each robot. Moreover, local supervisors are in charge of dynamically selecting the active tasks and their priority orders to properly perform the individual missions.

In the following, the definitions of the task functions and the detail on the supervisor are presented.

A. Tasks' definition

Just two tasks are sufficient to generate a flocking behavior in a group of robots when a rendez-vous point is assigned. An additional task is required in presence of obstacles. Each of the functions is defined for each robot and relies on the neighbors' information, only:

Lattice task. This task function $\sigma_l \in \mathbb{R}$ is aimed at keeping a constant distance (the lattice's scale) between the generic i -robot and one of its neighbors $\mathbf{p}_j \in \mathcal{N}_i$:

$$\sigma_l = \|\mathbf{p}_i - \mathbf{p}_j\| \quad \text{with} \quad \sigma_{l,d} = d. \quad (11)$$

Its Jacobian $\mathbf{J}_l \in \mathbb{R}^{1 \times 2}$ and Null $\mathbf{N}_l \in \mathbb{R}^{2 \times 2}$ matrices are defined as

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

where

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

Moving to rendez-vous task. The task function $\sigma_r \in \mathbb{R}^2$ is aimed to drive all the robots to converge to a meeting point; this facilitates the creation of a connected graph. Its definition is simply given by the robot position

$$\sigma_r = \mathbf{p}_i \quad \text{with} \quad \sigma_{r,d} = \mathbf{p}_{\text{RV}}, \quad (12)$$

where $\mathbf{p}_{\text{RV}} \in \mathbb{R}^2$ is the rendez-vous point. The (2×2) Jacobian is simply the Identity matrix and the Null space projector is the (2×2) null matrix.

Obstacle avoidance task. Obstacle avoidance for autonomous robots is a mandatory task and, resorting to the NSB approach, has been deeply discussed in previous papers such as, e.g., [6], [3]. Not surprisingly, the obstacle avoidance task function is formally equal to a lattice task:

$$\sigma_o = \|\mathbf{p}_i - \mathbf{p}_o\| \quad \text{with} \quad \sigma_{o,d} = d_o \quad (13)$$

and

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

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

It is worth noticing that each task function represents a single elementary behaviors for a single robot. In this sense, a task function can be used several times if needed. As an example, a robot can implement the lattice task with respect to several different neighboring robots laying in its interaction range. Further insight on the number and kind of tasks to be used is given in next subsection.

B. Supervisor

The supervisor is a higher level function that, for each robot, is in charge of selecting the priority of the active tasks. Referring to the considerations in [3] concerning the degrees of freedom (DOFs) of the tasks and those provided by the robot, it is convenient to properly take into account the dimension of each task and to avoid overspecifying the mission by assigning more constraints than the available DOFs; for instance, when fulfilment of the higher priority task needs all the DOFs of the robot, it results useless adding further tasks since the null-space of the higher priority task is empty and the lower priority tasks would not take effect.

Each robot is only aware of the other robots inside its interaction area; among them, a list containing robots $\in \mathcal{N}_i$ sorted by their distance with respect to i is considered, $k_i(1)$ being the closest. By referring to a two-dimensional case, each robot computes the desired velocities corresponding to the following elementary tasks:

- Lattice task with respect to the robot $k_i(1)$ (if at least one robot $\in \mathcal{N}_i$);
- Lattice task with respect to the robot $k_i(2)$ (if at least two robots $\in \mathcal{N}_i$);
- Moving to rendez-vous task (if present in the mission specification);
- Obstacle avoidance task (if $\|\mathbf{p}_i - \mathbf{p}_o\| < r$);

that 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 absence of obstacles in the set $\mathcal{N}_i$. In this case, the supervisor computes the Lattice tasks assigning the highest priority to the closest robot. Since the Lattice task is mono-dimensional ($\sigma_l \in \mathbb{R}$), if at least two robots belong to $\mathcal{N}_i$ the moving-to-rendez-vous task is discarded; otherwise, it is added as the lowest in priority. It is worth noticing that, even if more than two robots belong to $\mathcal{N}_i$, for the approach presented here it is sufficient to consider only the closest two and not all of them as for the approaches based on the use of potential functions.

Let us now consider one obstacle in the interaction range of the robot. The supervisor firstly computes the desired velocity without the obstacle; it then checks if the robot would approach the obstacle or go away from it. In the latter situation nothing is changed with respect to the non-obstacle case. If instead the robot would approach the obstacle, then the obstacle-avoidance task is selected as the primary task and all the other tasks are correspondingly lowered in priority; the last-priority one is eventually removed if the sum of the tasks' dimensions is larger than two.

For example, let us imagine a situation where the robot i has only one robot j inside the interaction range; its supervisor would then consider only the tasks in Table I. If, on the other hand, several robots and the obstacle are inside the interaction range, the supervisor would activate the tasks in II.

It is worth noticing that both in [21] and in this paper,

priority	task	dimension
1	lattice between i and j	1
2	rendez-vous	2

TABLE I
TASKS ACTIVATED BY THE ROBOT SUPERVISOR IN PRESENCE OF A UNIQUE NEIGHBOR.

priority	task	dimension
1	obstacle avoidance between i and $\mathbf{p}_o$	1
2	lattice between i and $k_i(1)$	1

TABLE II
TASKS ACTIVATED BY THE ROBOT SUPERVISOR IN PRESENCE OF MULTIPLE NEIGHBORS AND AN OBSTACLE

each robot needs to know the relative position to the other robots present in a set $\mathcal{N}_i$, that is in 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 tasks that take these into account.

V. SIMULATION RESULTS

To test the performance of the proposed control approach extensive simulations were run under SCILAB [26]. Following the results presented in [2], [1], in this paper we present the tests concerning the presence of static/moving rendez-vous points in a 2D implementation; the results were obtained running several simulations with different number of robots and different lattice's scales. Due to lack of space, a representative part of the simulations will be shown in this paper; some animations can be found at the URL <http://webuser.unicas.it/lai/robotica/video/> where also some experimental results obtained with a platoon of 7 wheeled mobile robots are available.

For all the simulations, the sampling time is set to $T = 0.1$ s, the CLIK gains are $\lambda_p = 0.5$, $\lambda_l = 1$ and $\lambda_o = 0.5$. At the start of the simulation the robots are randomly placed according to a Gaussian probability density function with null mean and variance proportional to the number of robots. It can be anticipated that in all the simulation runs the deviation energy (5) tends to zero with a rate that depends on several factors, among which the size of the group.

A. Two-dimensional case with static rendez-vous point

A team of 100 robots starting from a random configuration has to flock around a static rendez-vous point. Figure 2 shows the final configuration of the team after 20 s of numerical simulation. It is worth noticing that in this case an α -Lattice structure is asymptotically achieved. Moreover, the robots flock in a structure without fragmentation, that corresponds to a connected graph.

B. Two-dimensional case with moving rendez-vous point

Figure 3 shows 4 snapshots of a simulation with 40 robots; the rendez-vous point moves with a velocity $\dot{\mathbf{p}}_{rv} =$

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

$[2 \ 0]^T$ m/s. Flocking is asymptotically achieved while the formation follows the moving rendez-vous point. Notice that the group does not move as a solid object tracking the target, but rather propagates its motion through its members. This behavior is due to the local nature of the single robot controllers and the selected priorities for the tasks; the group, in fact, moves to fulfill the lattice task with a higher priority than the moving-to-rendez-vous task. It can also be observed that the group exhibits a sort of compression/expansion along the direction of rendez-vous movement.

C. Analysis of convergence time

The results presented above are only selected case studies of an extensive simulative analysis aimed at numerically testing the efficiency of the proposed approach under different conditions. One aspect that we have deeply investigated over several simulation runs is dependence of the convergence time from the size of the swarm. We report here the results of a study focused on 2D simulations with a static rendez-vous point in absence of obstacles; the same simulation parameters as those reported at the beginning of Section V are utilized.

Since the initial configuration is a critical parameter that influences the transient behavior of the swarm, 20 simulations with randomly deployed robots were run with teams of fixed number of robots. An uniform probability density function has been selected with ranges related to the square root of the swarm dimension to choose the initial displacements of the robots; this choice allowed 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 univocally detect when the flock reaches a steady state configuration. Several metrics have been proposed in the literature to give a mathematical measurement of how far is a configuration from an α -Lattice structure such as, e.g., the deviation energy defined in eq. (5), or the *social entropy* proposed by [16] and the *cohesion radius* proposed by [13]. In our study, the deviation energy has been used; in particular, the flocking is considered to reach a steady-state condition when the index is smaller than a given threshold.

Figure 4 shows a bar graph of the results; the abscissa reports the swarm dimension (the number of robots that compose the team), while for each assigned swarm dimension the ordinate reports the minimum, maximum and average convergence time for the 20 simulations. The darker columns refer to noiseless simulations while the lighter columns refer to simulations including noisy relative position measurements with uniform probability density function between ranges ± 20 cm; this value corresponds to the 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- α -Lattice configuration, in fact, a group of locally interacting robots is formed and the movement of one robot propagates through all the remaining ones; there is, thus, a long period of small adjustments among the robots before the α -Lattice configuration is reached. It can be observed that the presence of noise mainly has an adverse effect on the average convergence time of the flocking.

Fig. 4. Convergence time (minimum, average and maximum on 20 simulations) for an increasing number of robots. For each size, the darker column refers to a noiseless simulation while the lighter column refers to a simulation including noisy relative position measurements.

VI. CONCLUSIONS

The Null-Space-based Behavioral control approach has been applied to a wide range of robotic systems in recent years. In the case of multi-agent systems, problems such as, e.g., formation control, escorting a target, or reconfiguration of a mobile ad-hoc network, have been successfully achieved.

In this paper, the flocking problem of a group of dynamic agents has been addressed. Flocking in presence of a static or moving rendez-vous point has been discussed and verified by numerical simulations; moreover, flocking performance execution changing the size of the team and adding measurement noise has been analyzed. It has been shown that very simple task functions activated by a supervisor can successfully handle the flocking problem. Further research will focus on bringing the approach to experiments.

REFERENCES

- [1] G. Antonelli, F. Arrichiello, and S. Chiaverini. Flocking for multi-robot systems via the null-space-based behavioral control. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1409–1414, Nice, France, Sept. 2008.

Fig. 3. Snapshots for 40 robots flocking around a moving rendez-vous point and without obstacles. The dots represent the robots, the solid lines the connections. The rendez-vous point, represented by the cross, moves from $[0 \ 0]$ m to $[60 \ 0]$ m in 20s.

- [2] G. Antonelli, F. Arrichiello, and S. Chiaverini. The NSB control for 3-dimensional flocking of multi-robot systems. In *2008 IEEE International Conference on Information and Automation*, pages 101–106, Hunan, China, June 2008.
- [3] 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.
- [4] 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, Mexico, Dec. 2008.
- [5] G. Antonelli and S. Chiaverini. Kinematic control of a platoon of autonomous vehicles. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, TW, Sept. 2003.
- [6] G. Antonelli and S. Chiaverini. Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, 22(6):1285–1292, Dec. 2006.
- [7] R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.
- [8] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.
- [9] Y. Cao, A.S. Fukunaga, and A.B. Kanhg. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.
- [10] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.
- [11] S. Chiaverini, G. Oriolo, and I. D. Walker. *Springer Handbook of Robotics*, chapter Kinematically Redundant Manipulators, pages 245–268. B. Siciliano, O. Khatib, (Eds.), Springer-Verlag, Heidelberg, D, 2008.
- [12] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.
- [13] D. Gu and H. Hu. Using fuzzy logic to design separation function in flocking algorithms. *IEEE Transactions on Fuzzy Systems*, 16(4):826–838, August 2008.
- [14] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
- [15] M. Ji and M. Egerstedt. Distributed Coordination Control of Multiagent Systems While Preserving Connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.
- [16] M. Kumar, D. Milutinović, and D. P. Garg. Role of stochasticity in self-organizing of robotic swarms. In *Proceedings 2008 American Control Conference*, pages 123–128, Seattle, June 2008.
- [17] Y. Li and X. Chen. Modeling and simulation of swarms for collecting objects. *Robotica*, 24(03):315–324, 2005.
- [18] N. Mansard and F. Chaumette. Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics and Automation*, 23(1):60–72, 2007.
- [19] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75–88, Aug. 2007.
- [20] M.J. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2):321–331, 1995.
- [21] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, March 2006.
- [22] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan. 2007.
- [23] W. Ren and R.W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communications and Control Engineering. Springer, Berlin, 2008.
- [24] W. Ren, R.W. Beard, and E.M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, Apr. 2007.
- [25] C. Reynolds. Flocks, herd and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.
- [26] Scilab-URL. <http://www.scilab.org/>. INRIA ENPC.