Hybrid Flock - Formation Control Algorithms

Cyrill Baumann, Jonas Perolini, Emna Tourki, and Alcherio Martinoli

Distributed Intelligent Systems and Algorithms Laboratory
School of Architecture, Civil and Environmental Engineering,
École Polytechnique Fédérale de Lausanne, Switzerland,
firstname.lastname@epfl.ch

Abstract. Two prominent categories for achieving coordinated multi-robot displacement are flocking and navigation in formation. Both categories have their own body of literature and characteristics, including their respective advantages and disadvantages. While typically, they are treated separately, we believe that a combination of flock and formation control represents a promising algorithmic solution. Such an algorithm could leverage a combination of characteristics of both categories best suited for a given situation. In this work, we therefore propose two distributed algorithms, able to gradually and reversibly shift between flocking and formation behaviors using a single parameter $\mathcal{W}$. We evaluate them using both simulated and real robots with and without the presence of obstacles. We find that both algorithms successfully trade off flock density for formation error. Furthermore, using a funnel experiment as application case study, we demonstrate that an adaptive shift between flock and formation behavior, using a simple method to define $\mathcal{W}$ in real-time using exclusively on-board resources, results in a statistically relevant reduction of traversing time in comparison to a non-adaptive formation control algorithm.

Keywords: flocking, formation control, multi-robot coordination

1 Introduction

In recent years, Multi-Robot Systems (MRS) have gained importance, as coordinated teams are able to perform tasks which would be unfeasible for single robots. Additionally, their inherent redundancy introduces robustness to the system allowing it to successfully complete the overall mission objectives even when some individual agents fail. Spatial coordination of MRS is crucial in a wide range of applications, ranging from search and rescue missions over surveillance or escorting missions to spacecraft formations. Existing algorithms that tackle this challenge can be split into two main categories: those that manage tight spatial topologies, most commonly referred to as formation control, and those that manage loose ones, usually called flocking.

Flocking has its origin in nature, where such behavior achieves multiple functionalities (e.g., energy benefits, navigation accuracy, protection from predators) and is present in all media (e.g., schools of fish and flocks of birds). This

behavior has been first engineered in [1] for the purpose of modeling the behavior of birds, fish, etc. in computer animation. It has since been improved upon, but the main principles stayed the same. The behavior of each individual agent can be expressed as the sum of three different behaviors, as defined by Craig Reynolds:

- Collision avoidance: avoid collisions with nearby flockmates
- Velocity matching: attempt to match velocity with nearby flockmates
- Flock centering: attempt to stay close to nearby flockmates

The result is a swarm-like behavior with (in steady state) almost equal distances among the flock members but without any precise formation. Additionally, formal methods for designing flocking behavior have also been proposed, for example in [2], where three different kinds of lattices are leveraged to form triangle-like flocks, allowing for splitting and rejoining as well as squeezing maneuvers to avoid obstacles.

Formation control is an actively studied topic with a large amount of prior work, including recent surveys such as [3]. There exist three main approaches for addressing formation control: *Leader-Follower*, *Virtual Structure (VS)* and *Behavior-Based (BB)* methods. In the **LF** approach, one agent is designated as leader. All other agents, called followers, are then tasked to keep a predefined relative position to the leader, resulting in formation shape retention [4]. The **VS** approach [5], consists of three steps. First the virtual structure (corresponding to the predefined formation) is aligned to the current positions of the agents in order to minimize the distance from every agent to its corresponding spot within the formation. Then the virtual structure is moved in the desired direction of movement. In the third step, the agents are controlled towards their corresponding positions in the shifted virtual structure. The **BB** approach is based on defining several desired behaviors for every agent. The control is then either a weighted average of the resulting motion vectors created by the individual behaviors [6], a Finite State Machine (FSM) switching between them [7] or a more elaborate combination of behaviors [8]. A BB approach results in a collective response fairly adaptive to different environmental situations. While this represents the main advantage of this approach, one major drawback is that the resulting spatial topology emerges from local interactions and is therefore difficult to control it precisely.

In multiple related works the formation control has subsequently been "loosened" up, usually to account for obstacle avoidance. In [9] the formation is capable of splitting and rejoining. In [10] and [11] dynamical formation changes are leveraged. In [12] a potential field augmented with formation transforming is used. Deformable formations are also leveraged in [13].

On the flocking side, it is worth noting that, in comparison to the flocking based on Reynold's rules, flocking based on lattices (as in [2]) can be considered significantly more rigid and thus closer to formation control.

However, above approaches all remain faithful to their respective category of algorithms. More specifically, to the best of our knowledge, no work exists which is capable of shifting gradually from flocking to formation control or vice-versa.

Yet this gradual shift would enable a smooth transition between the characteristics of both types of algorithm with an infinite number of intermediate states in between. Such a continuous change could then be leveraged in numerous scenarios. For example, it could be used to optimize multi-robot navigation by trading off formation precision versus energy consumption needed to maintain it.

In this work, we propose and experimentally validate two such algorithms, designed to gradually shift between a flock and a formation behavior. For both algorithms proposed, the formation control is based on a graph-based, Laplacian control algorithm [14], which can be seen as part of the VS category. The corresponding flocking algorithm used differs between the two proposed solutions: **Pot2Lapl** leverages a potential based flocking algorithm based on [2], while **Flock2Lapl** uses the original Reynolds rules ([1]), more specifically the version introduced in [15].

The remainder of this paper is structured as follows: we start by introducing the two algorithms Flock2Lapl and Pot2Lapl. We then present our two experimental evaluation setups, report and discuss the algorithms' performances, before ending with some conclusive remarks.

2 Methodology

We consider a homogeneous group of N differential wheeled robots, tasked to follow a predefined trajectory while forming and maintaining spatial coordination among them. This spatial coordination varies from a flock to formation control depending on the scalar $\mathcal{W}$.

The robot control is designed following the framework introduced and described in detail in [16]. The control algorithm consists of the following seven functional elements:

1. Pose estimation: each robot is equipped both with a local range-and-bearing system, Global Navigation Satellite System (GNSS) capabilities and global communication which is used to transmit (and receive) position data. This effectively results in a global, occlusion-less sensing of neighboring robots. All information is fused using an Extended Kalman Filter.
2. Consensus on orientation: formation orientation is not established a priori but obtained online using Laplacian consensus.
3. Position allocation: a Hungarian algorithm [17] is used to assign each robot the optimal position within the formation (minimizing the overall distances between the robots and their respective target positions).
4. Coordination algorithm: the estimated relative positions of neighboring robots (X_i with $i \in [1, N]$) is used for both coordination algorithms. They are specified in Sections 2.1 and 2.2 respectively.
5. Group movement: the common goal is given as a temporal trajectory in a global reference frame.
6. Collision avoidance: the commonly used Braitenberg algorithm is applied using the modifications introduced in [16] to cast it in the motion vector framework.

7. Robot control: a single integrator control law including a damping term as in [16] is used.

It is further worth noting that the motion vectors of elements 4, 5 and 6 ($\mathcal{X}_{coord}$, $\mathcal{X}_{traj}$ and $\mathcal{X}_{col}$ respectively) are summed to obtain the final motion vector.

Both coordination algorithms, *Pot2Lapl* and *Flock2Lapl* use as formation control part of their respective algorithms a graph-based, Laplacian control algorithm. Using the standard feedback method for a single integrator kinematic model [18], we obtain for robot j:

$$\dot{z}_j = \frac{1}{N} \sum_{i=1}^{N} \Big(- \mathcal{L}_{i,j} z_{i,j} + b_{i,j} \Big) \tag{1}$$

with $\mathcal{L}$ the Laplacian matrix based on the underlying connectivity graph (which is fully connected in this work), $z_{i,j}$ the position of robot i and $b_{i,j}$ the desired position of robot i, both defined in the local coordinates of robot j. $b_{i,j}$ can be calculated from the desired positions in global coordinates $\mathcal{B}_N$ for a given number of robots N. Following the reasoning in [16], we can express the motion vector of the Laplacian control algorithm as:

$$\mathcal{X}_j^{lapl} = -\dot{z}_j dt \tag{2}$$

with $\mathcal{B}_N$, dt numerical parameters. Numeric values used in this work are summarized in Table 1. The resulting formation for $N = 5$ is depicted in Figure 3b.

2.1 Pot2Lapl

Intuitively, Pot2Lapl corresponds to a potential-based flocking algorithm (c.f., [2]), with an additional attraction towards holonomic, massless virtual agents following Laplacian control. Then, varying a single attractive weight $\mathcal{W}$ towards the virtual agents allows to gradually shift from pure flocking (i.e. virtual agents are ignored) to pure formation control (i.e. attraction to virtual agents dominates flocking).

More specifically, the target motion vector $\mathcal{X}_{coord}$ is given by

$$\mathcal{X}_j^{coord} = \sum_{i=1}^{N} \Big(\frac{d\psi}{dz}(||z_{i,j}|| - d_{ref}) \frac{z_{i,j}}{||z_{i,j}||} \Big) + \mathcal{W} \mathcal{X}_j^{virt} \mid i \neq j \tag{3}$$

With ψ the potential function as defined below, d_{ref} the desired inter-robot distance and $\mathcal{X}_j^{virt}$ the position of the virtual agent associated to robot j following a Laplacian-based control. That is $\mathcal{X}_j^{virt} = \mathcal{X}_j^{lapl}$.

To define ψ, we first define a base potential function $\psi_{base}(z)$ similar to [2]:

$$\psi_{base}(z) = \frac{a+b}{2} \Big(\sqrt{1 + (z+c)^2} - \sqrt{1 + c^2} \Big) + \frac{a-b}{2} z \tag{4}$$

using $c = \frac{|a-b|}{2\sqrt{ab}}$ and a,b numerical parameters.

Different from the work in [2], we further define a custom cut-off function $\rho(z)$, to limit the influence of far-away neighbors:

$$\rho(z) = \begin{cases} 0, & \text{if } z < \frac{d_{ref}}{r} + \delta \\ 1 - \frac{1}{2}\left(1 + cos\left(\pi \frac{z - \frac{d_{ref}}{r} - \delta}{1 - \frac{d_{ref}}{r} - \delta}\right)\right), & \text{if } \frac{d_{ref}}{r} + \delta \leq z \leq 1 \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

Additionally, we define a piece-wise constant function $B(z)$ to increase the repulsive effect of the potential function:

$$B(z) = \begin{cases} K_{rep}, & \text{if } z < 0 \\ K_{att}, & \text{otherwise} \end{cases} \tag{6}$$

We can then define a piece-wise derivable, cut-off potential function ψ

$$\psi(z) = B(z)\left(\psi_{base}(z) - \rho\left(\frac{z + d_{ref}}{r_{cut}}\right)\left(\psi_{base}(z) - \psi_{base}(r_{cut})\right)\right) \tag{7}$$

with d_{ref}, δ, K_{rep}, K_{att} and r_{cut} numerical parameters. The proposed cut-off potential function has the advantage to have a unique global minima at the desired robot inter-distance, solving the multiple global minima drawback from the one in [2]. Figure 1 shows the resulting $\psi(z - d_{ref})$ and the corresponding gradient $\frac{d\psi}{dz}(z - d_{ref})$ with the numerical parameters used in this work.

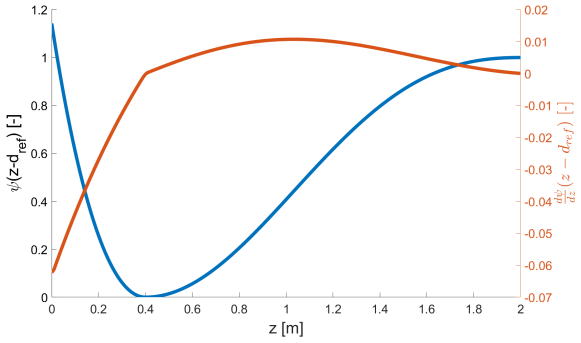


Fig. 1: Potential function ψ it's corresponding gradient $\frac{d\psi}{dz}(z - d_{ref})$ with the parameters used in this work as function of the inter robot distance z.

2.2 Flock2Lapl

Flock2Lapl is based on the original Renyolds rules [1], using the formulations of [15]. Intuitively, Flock2Lapl is an extension of the three rules to include a fourth rule: attraction to a virtual agent following a Laplacian formation control. Formally, we can write the target motion vector $\mathcal{X}_{coord}$ of robot j as

$$\mathcal{X}_j^{coord} = K_{coh} X_j^{coh} + K_{sep} X_j^{sep} + \mathcal{W} X_j^{virt} \tag{8}$$

using again $\mathcal{X}_j^{virt} = \mathcal{X}_j^{lapl}$, with the cohesion being defined as

$$X_j^{coh} = \frac{1}{N} \sum_{i=1}^{N} z_{i,j} \tag{9}$$

and the separation vector defined as

$$X_j^{sep} = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} \frac{-z_{i,j}}{||z_{i,j}||^2}, & \text{if } ||z_{i,j}|| < s \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

with K_{coh}, K_{sep} and s numerical parameter.

Table 1: Numerical parameters used in this work, listed in order of their appearance.

Parameter	**Value**
$\mathcal{B}_5$	{(0, 0), (0,1.2), (0.4, 0.3), (0.4, 0.9), (0.8, 0.6)} [m]
$\mathcal{B}_{10}$	{(0.8, 0), (0.8, 0.4), (0.8, 0.8), (0.8,1.2), (1.2, 0.2), (1.2, 0.6), (1.2,1), (1.6, 0.4), (1.6, 0.8), (2, 0.6)} [m]
dt	50 [ms]
a	1 [-]
b	7 [-]
d_{ref}	0.4 [-]
δ	0 [m]
r_{cut}	2 [m]
K_{rep}	10 [-]
K_{att}	0.9 [-]
K_{coh}	1 [-]
K_{sep}	0.15 [-]
s	0.45 [m]
r_s	0.3 [m]
$\mathcal{W}_0$	4 [-]

3 Experimental evaluation

This section introduces the robotic platform used to demonstrate the algorithms, the experimental setup for both simulated and real robot experiments, as well as the evaluation metrics used.

3.1 Robotic platform

All experiments were conducted using both simulated and real Khepera IV robots [19]. Khepera IV robots are differentially driven vehicles with a diameter of 14 cm and a maximal speed of $\sim$ 81 cm/s. They are further equipped with eight infrared (IR) proximity sensors with a range of 20 cm used for obstacle avoidance. As illustrated in Figure 2, the robots are enhanced with a custom range-and-bearing module [20] for relative localization and an active marker module featuring two LEDs which enables accurate tracking through the SwisTrack software [21]. Communication between the robots is leveraging UDP/IP and is assumed to be error-free and to have a range larger than the experimental arena.

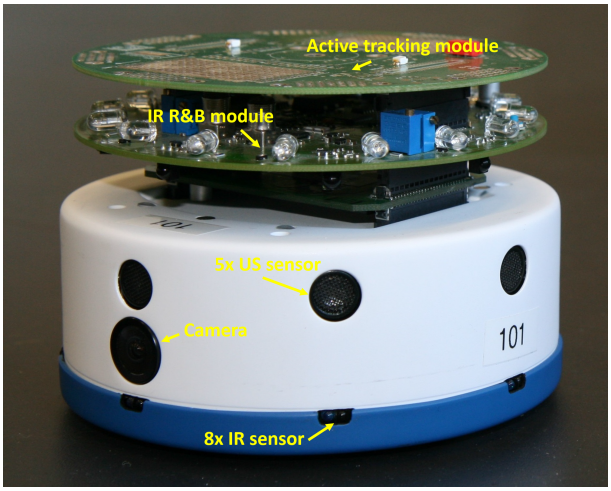


Fig. 2: Khepera IV robot with a custom range-and-bearing module as well as tracking module. In this work, both the camera and the US sensors are not used and thus not further mentioned.

3.2 Evaluation metrics

The performance of the coordination algorithms are evaluated using four metrics for every temporal instant t given the robots' positions Z_i with $i \in N$ in a global reference frame.

– $\Phi_{density}$ evaluates the density of the group with respect to an ideal, lattice-like flock with d_{ref} between all robots. Based on circle packing, it compares A_r, the enclosing circle of all robots with A_o, the optimal enclosing circle for the given number of robots.

$$\Phi_{density} = \frac{1}{1 + ||A_o - A_r||} \tag{11}$$

– $\Phi_{formation}$ captures the error between the robots' positions and their respective desired positions Z_{D,Z_F} within the formation centered at Z_F using the optimal position assignment within the formation.

$$\Phi_{formation} = \underset{Z_F}{\operatorname{argmin}} \left(\frac{1}{1 + \left(\sum ||Z - Z_{D,Z_F}|| \right)} \right) \tag{12}$$

– Φ_{order} (adopted from [15]) evaluates the heading alignment of the robots:

$$\Phi_{order} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{\dot{Z}_i \dot{Z}_j}{||\dot{Z}_i|| \, ||\dot{Z}_j||} \tag{13}$$

– Φ_{safety} (adopted from [15]) measures the risk of collision among robots:

$$\Phi_{safety} = 1 - \frac{n_s}{N(N-1)} \tag{14}$$

with n_s the number of occurrences where two robots are "too close", that is closer than r_s, a numerical parameter: $n_s = |(i,j)\, s.t.\, j \neq i \;\; \& \;\; ||Z_i - Z_j|| < r_s|$

We further denote $\overline{\Phi}$ the respective average performances over the complete experiment duration.

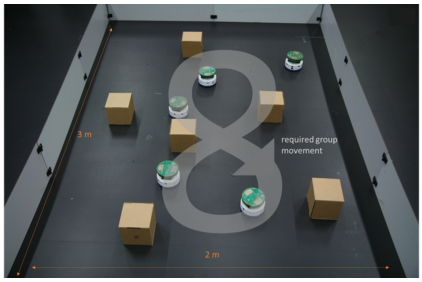
3.3 Experimental setup

Webots [22], an open-source, high-fidelity robotics simulation platform has been used for the simulated experiments. All simulated sensors and actuators of the simulated robots were calibrated to match those of the real robots.

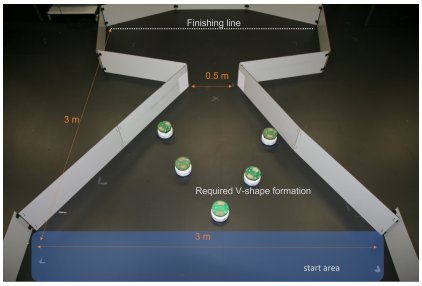
For the physical experiments, SwisTrack [21] is used in combination with a GigE color camera to both track the robots in order to calculate the evaluation metrics, as well as to emulate absolute localization such as that provided by a GNSS module in indoor settings. SwisTrack is calibrated using the algorithm proposed by Tsai et al. [23] which results in a spatial error of 1.06 +/- 0.65 cm and a negligible temporal error of less than 1 mm amplitude. Emulated GNSS information is sent to the robots via UDP/IP at a rate of 1 Hz.

3.4 Characterization experiment

Characterization experiments are conducted similarly in simulation and reality: five Khepera IV robots are randomly placed in an 3 x 2 m^2 arena, as depicted in Figure 3a. Using the algorithmic framework described in Section 2, both algorithms are then evaluated for different parameters, as summarized in Table 2. An 8-shaped trajectory has been chosen as group movement for all experiments. For the experiments with obstacles, six square obstacles of 20 x 20 cm are placed arbitrarily in the arena. We note that the positions of the obstacles are kept constant over all experiments to increase comparability among them. Figure 3a shows the corresponding obstacle positions used for both real and simulated experiments.



(a) Five randomly placed robots in the real 3 x 2 m arena populated with six obstacles.



(b) Five robots in the funnel crossing scenario arena. Their formation is already slightly deformed on the right side due to the wall.

Fig. 3: Experimental arenas used in this work. The required V-shaped formation is identical for both sets of experiments.

Table 2: Experimental conditions evaluated in this work

Environment	$\mathcal{W}$	N
Webots without obstacles	[0, 1, 2, 5]	[5, 10]
Webots with obstacles	[0, 1, 2, 5]	5
Real robots	[0, 1, 2, 5]	5

3.5 Funnel experiment

A funnel experiment, akin to the one in [16], is used as case study to demonstrate the performance of the algorithms. Five robots are placed randomly within a starting zone (see Figure 3b). The robots are then tasked to go through the funnel and pass the finish line, while maintaining the spatial coordination. The migration urge is given as temporal trajectory in the global reference frame. Real robot experiments were repeated 10 times, simulated ones 20 times. In addition to the four metrics introduced previously, we record for every experimental run the time taken until the last robot crosses the finishing line.

In contrast to the characterization experiments of Section 3.4, $\mathcal{W}$ is calculated dynamically on-board for every robot and timestep: $\mathcal{W} = max(\mathcal{W}_0 - N_{obst-detect}, 0)$ with $\mathcal{W}_0$ a numerical parameter corresponding to the desired "rigidness" of the formation and $N_{obst-detect}$ the number of infrared sensors detecting an obstacle. The performance of this "adaptive" algorithm is compared to a "non-adaptive" version which uses $\mathcal{W} = \mathcal{W}_0$.

4 Results and discussion

Figure 4 shows the resulting performance of the Flock2Lapl algorithm on the four metrics presented before for different $\mathcal{W}$. As intended, with increasing $\mathcal{W}$,

the ϕ_{form} increases and $\phi_{density}$ decreases. It is further worth noting that ϕ_{safety} and ϕ_{order} are not impacted by $\mathcal{W}$.

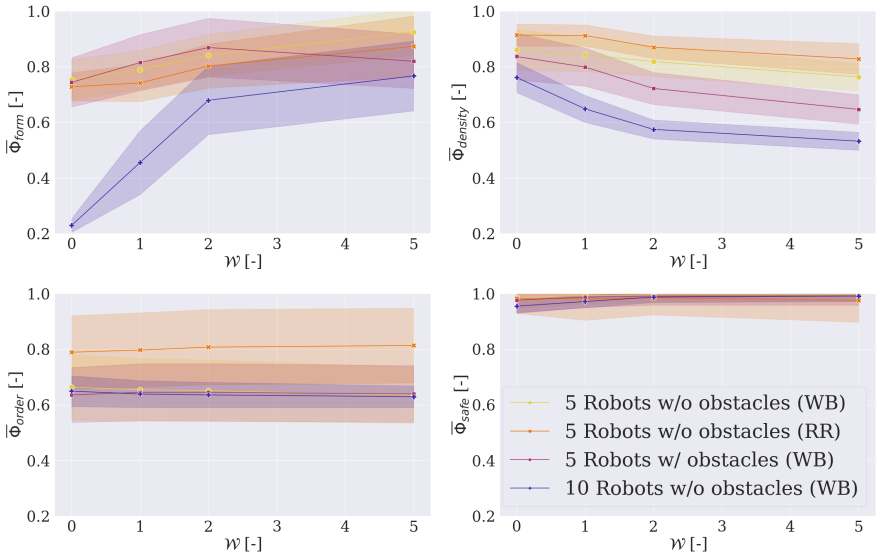


Fig. 4: Experimental results for the Flock2Lapl algorithm, using both 5 and 10 robots in simulation without obstacles (WB), using 5 robots in simulation with obstacles (WB) and using 5 robots in reality without obstacles (RR).

Figure 5 shows the corresponding performances for the Pot2Lapl algorithm which are similar to the results of Flock2Lapl. Figures 4 and 5 further show the resulting performance of both algorithms for 10 robots. We note that both ϕ_{form} and $\phi_{density}$ decrease with increasing N. While this is undesired, we attribute this performance to the specific evaluation metrics chosen for this comparison. Indeed, both metrics are not normalized with respect to the number of robots, explaining the results obtained.

Figure 6 shows the results of the funnel experiments. For all metrics, a one-way ANOVA revealed that there was a statistically significant difference in the respective metric between at least two groups. However, Tukey's HSD Test for multiple comparisons found that the mean value of the metrics differs only between simulated and real experiments, with following exceptions: for Flock2Lapl, ϕ_{form} is statistically different for simulated experiments ($p < 0.05$), for Pot2Lapl, $\phi_{density}$ is statistically different for real experiments ($p < 0.01$), for both algorithms the traversing time is statistically different for both simulated and real experiments ($p < 0.05$). This indicates that our proposed algorithms successfully decreased traversing time, demonstrating one possible use case. We note that the impact on ϕ_{form} is limited by the averaging over the whole experiment.

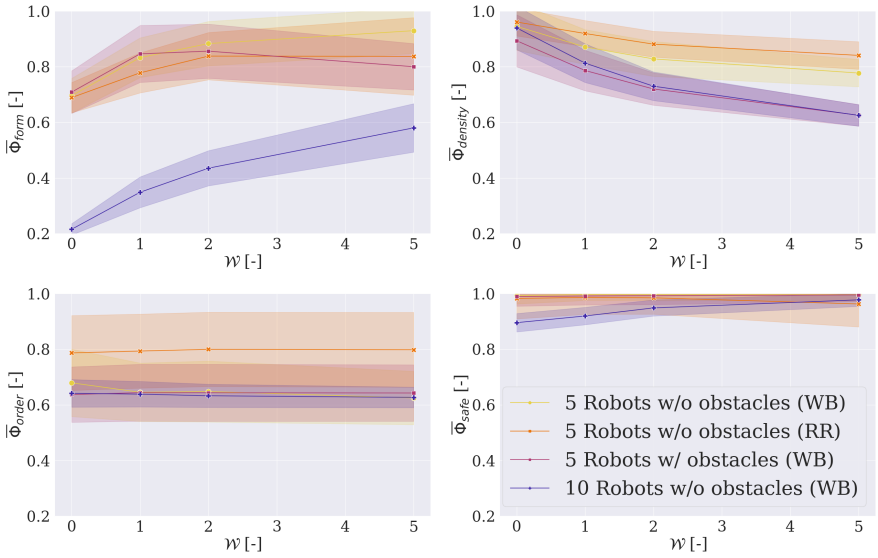


Fig. 5: Experimental results for the Pot2Lapl algorithm, using both 5 and 10 robots in simulation without obstacles (WB), using 5 robots in simulation with obstacles (WB) and using 5 robots in reality without obstacles (RR).

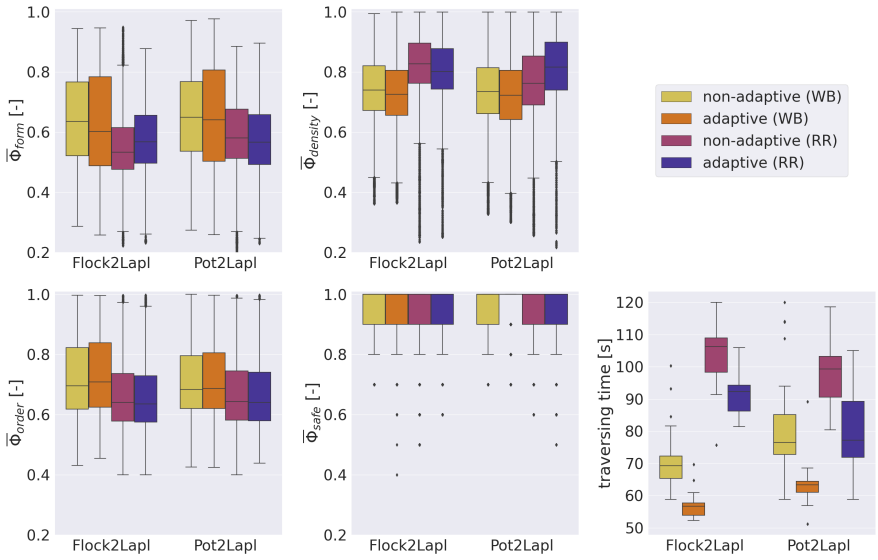


Fig. 6: Experimental results for the funnel experiment using five robots in simulation (WB) and reality (RR).

However, given a more sensitive metric, we would expect this to be a trade-off between traversing time and ϕ_{form}.

We further acknowledge that, as this work focuses on the coordination algorithms, the sensing capabilities of the robots are maximized. Naturally, further work will be required to analyze the performances of the algorithms under more realistic situations considering exclusively local sensing affected by occlusions.

5 Conclusion

In this work we introduced two hybrid flock - formation control algorithms. By varying a single parameter $\mathcal{W}$, both algorithms are able to change gradually from a pure flocking behavior to a formation behavior and vice-versa. We then characterized the behavior of the two algorithms using four dedicated metrics for different numerical values of $\mathcal{W}$ in both simulated and real robot experiments.

We further introduced a simple method to compute $\mathcal{W}$ on-board, based on the number of infrared sensors detecting an obstacle. By comparing this adaptive algorithm to a non-adaptive formation control algorithm for a funnel scenario, we find that a significant traversing time reduction is achieved by the adaptive algorithm.

Acknowledgements

We would like to thank Masaki Haruna, Masahiko Kurishige and Tomoki Emmei of the Advanced Technology R&D Center, Mitsubishi Electric Corporation, Amagasaki City, Japan for the fruitful technical discussions all along this work. This work is financially supported by Mitsubishi Electric Corporation, Japan.

References

1. C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," in *ACM Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34, 1987.
2. R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, 2006.
3. Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.
4. P. K. C. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation," *Journal of Robotic Systems*, vol. 8, no. 2, pp. 177–195, 1991.
5. K.-H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 132–139, 1996.
6. T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.
7. H. Wang and M. Rubenstein, "Shape Formation in Homogeneous Swarms Using Local Task Swapping," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597–612, 2020.

8. G. Antonelli, F. Arrichiello, and S. Chiaverini, "The NSB control: a behavior-based approach for multi-robot systems," *Paladyn, Journal of Behavioral Robotics*, vol. 1, pp. 48–56, 2010.
9. S. Monteiro and E. Bicho, "Attractor dynamics approach to formation control: Theory and application," *Auton. Robots*, vol. 29, pp. 331–355, 2010.
10. J. M. Vilca Ventura, L. Adouane, and Y. Mezouar, "Adaptive Leader-Follower Formation in Cluttered Environment Using Dynamic Target Reconfiguration," in *12th International Symposium on Distributed Autonomous Robotic Systems*, pp. 237–254, 2014.
11. A. Wasik, J. N. Pereira, R. Ventura, P. U. Lima, and A. Martinoli, "Graph-based distributed control for adaptive multi-robot patrolling through local formation transformation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1721–1728, 2016.
12. W. L. Seng, J. C. Barca, and Y. A. Şekercioğlu, "Distributed formation control of networked mobile robots in environments with obstacles," *Robotica*, vol. 34, no. 6, pp. 1403–1415, 2016.
13. J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "A distributed formation-based odor source localization algorithm - design, implementation, and wind tunnel evaluation," in *IEEE International Conference on Robotics and Automation*, pp. 1830–1836, 2015.
14. R. Falconi, S. Gowal, J. Pugh, and A. Martinoli, "Graph-based distributed control for non-holonomic vehicles engaged in a reconfiguration task using local positioning information," in *Second Int. Conf. on Robot Communication and Coordination*, 2009.
15. E. Soria, F. Schiano, and D. Floreano, "The Influence of Limited Visual Sensing on the Reynolds Flocking Algorithm," in *Third IEEE International Conference on Robotic Computing*, pp. 138–145, 2019.
16. C. Baumann and A. Martinoli, "A Modular Functional Framework for the Design and Evaluation of Multi-Robot Navigation," *Robotics and Autonomous Systems*, vol. 144, p. 103849, 2021.
17. H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
18. M. Mesbahi and M. Egerstedt, "Graph Theoretic Methods in Multiagent Networks," in *Princeton Series in Applied Mathematics*, 2010.
19. J. M. Soares, I. Navarro, and A. Martinoli, "The Khepera IV mobile robot: performance evaluation, sensory data and software toolbox," in *Second Iberian Robotics Conference*, vol. 417, pp. 767–781, 2016.
20. J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli, "A Fast Onboard Relative Positioning Module for Multirobot Systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 151–162, 2009.
21. T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "SwisTrack - a flexible open source tracking software for multi-agent systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4004–4010, 2008.
22. O. Michel, "WebotsTM: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, pp. 39–42, 2004.
23. R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.