Toward Emerging Cubic-Spline Patterns With a Mobile Robotics Swarm System

Belkacem Khaldi, Fouzi Harrou, *Member, IEEE*, Foudil Cherif, and Ying Sun

***Abstract*—An innovative and flexible approach is introduced to address the challenge of self-organizing a group of mobile robots into cubic-spline-based patterns without any requirement of control points. Besides the self-organization of mobile robots, the approach incorporates a potential field-based control for obstacle/collision avoidance. This will offer more flexibility to swarm robots to efficiently deal with many practical situations, including smoothly avoiding obstacles during movement or exploring and covering areas with complex curved patterns. Essentially, this challenge is approached by proposing a formation control model based on a smoothed particle hydrodynamic estimation technique, which uses special cubic-spline kernel functions applied here to interpolate the density of each robot in the swarm. The moving information is used to weigh the distances to the robot's neighbors available in its field of view. Then, an artificial physics mesh is finally built among each robot and its three available neighbors having the smallest weighted distances. Significant results toward emerging cubic-spline patterns are shown with a swarm of foot-bot mobile robots simulated in the ARGoS platform. Analysis results with different metrics are also conducted to assess the performance of the model with different swarm sizes and in the presence of sensory noise as well in the presence of partially faulty robots.**

***Index Terms*—ARGoS, artificial physics mesh, self-organized cubic-spline patterns, smoothed particle hydrodynamic (SPH), swarm robotics.**

I. Introduction

SELF-ORGANIZING patterning is one of the astonishing natural phenomena, which is often observed in a number of ecosystems, such as flocks of birds and herd mammals, and seashells and fish schooling [1], or in the biological process of self-organizing cells into organs [2]. What characterizes this type of pattern is that it only emerges from the application of local simple interaction rules between the swarm members in a fully distributed and decentralized manner. Biologists refer this to self-organization—a spontaneous process, where certain forms of sudden global orders emerge without external controls [3]. This process and some of its key properties, such as scalability, flexibility, and robustness have become one of the main sources of inspiration in developing robotics swarm systems [4], [5].

Recent advances in this sort of robotics systems make it possible to design a large scale of cheap autonomous simple robots collaborating together to perform various tasks beyond the capability of any individual robot. Nowadays, swarm robotic systems are getting to be a reality with application extending from environmental exploration to nanomedicine [4], [6].

Pattern formation in such robotics systems is one of the challenging problems that has been widely discussed in the literature. It can be presented as the way in which certain members of the swarm coordinate their efforts to form a certain form and maintain it in seek to accomplish a complex task [5]. In the majority of cases, the shape to be formed could be a predefined form obtained by following a predefined trajectory or an adaptive form produced in a self-organized manner by following simple local interaction rules [7]. Control algorithms of the latter type of formations have attracted much attention since they allow robots to be more adaptable to unknown and changing environments. The existing works on this type of control are generally inspired by flocking aggregation patterns and spatial patterning [7].

While several studies have been conducted to tackle the problem of flocking-based patterns in the context of multirobotics and swarm robotics (see a detailed review in [7]). However, most of them approached the problem by focusing only on the study of the coordination movement as well as on the cohesion of the swarm. Also, even some studies have put particular emphasis on investigating the role of information transfer (e.g., sharing orientation information) in understanding the flocking process [8]–[10]. However, very few of them have specifically addressed the spontaneous patterns emerging from their proposed models. Furthermore, none of them has tackled the emergence of cubic-spline patterns in such robotic systems through which we believe can be very useful in addressing the following scenarios.

1) In obstacle avoidance, a swarm of robots can move in a cubic-spline formation to smoothly avoid obstacles.

Manuscript received May 12, 2020; revised December 20, 2020; accepted January 23, 2021. Date of publication January 27, 2021; date of current version June 10, 2022. This work was supported in part by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award OSR-2019-CRG7-3800; in part by the École Supérieure en Informatique 08 Mai 1945, Sidi Bel Abbés, Algeria; and in part by the LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, Biskra, Algeria. *(Corresponding author: Belkacem Khaldi.)*

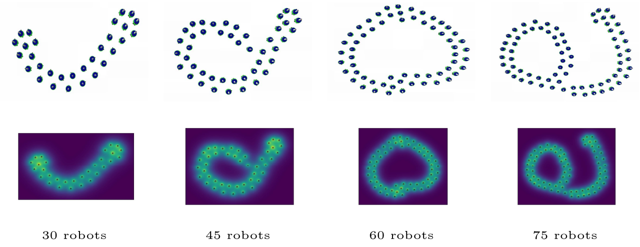
Belkacem Khaldi is with the Laboratoire de Recherche en Informatique (LabRI-SBA) de lŠESI-SBA, Ecole Supèrieure en Informatique de Sidi Bel Abbès, 22000 Yvette, Algeria (e-mail: b.khaldi@esi-sba.dz).

Fouzi Harrou and Ying Sun are with the Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia (e-mail: fouzi.harrou@kaust.edu.sa).

Foudil Cherif is with the LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, Biskra 07000, Algeria.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCDS.2021.3054997.

Digital Object Identifier 10.1109/TCDS.2021.3054997



30 robots 45 robots 60 robots 75 robots

Fig. 1. Few of self-organized cubic-spline patterns obtained from different swarm sizes performing our approach: swarm of simulated foot-bot robots (upper) with the corresponding swarm density map (bottom).

2) In exploration and coverage of areas, a group of robots may form cubic-spline patterns to cover areas with complex curved zones.
3) In artistic shows, a set of drones can be used to create cubic-spline art designs in the sky for entertainment or sporting events.
4) In education, a group of miniature robots can be implied in creating letters, numbers, or simple shapes based on cubic-spline patterns for educational purposes for children.

To address this issue, we propose here a control model for self-organizing patterning with a robotics swarm system. The main contributions of this article are as follows.

1) The proposition and implementation of a self-organized formation control model that is particularly able to emerge cubic-spline-based shapes (see Fig. 1 as examples) without the need of any control points.
2) A neighboring selection strategy for each robot that is restricted to the three available neighbors having the smallest weighted distances.
3) The introduction of a modified performance metric in assessing the performance of the proposed formation control model.

The proposed model (see Section III-A) is mainly inspired by the observation of physics and, therefore, follows an artificial physical approach. The model incorporates a technique for estimating the density of a robot in the swarm using a smoothed particle hydrodynamic (SPH) estimation technique [11] (see Section III-A). Note that based on a set of kernel interpolation functions, the SPH technique can approximate physical quantities moving with particles [12]. To this end, we use the M_4 cubic-spline kernel functions to approximate the density of the robot as physical information moving with it [11]. This information will then be used to weigh the distances to the neighbors of each robot. The weighted distances must then be sorted in ascending order and each robot will then apply an artificial viscoelastic physics law as the main attractive model among the robot's topological neighbors. Note that to obtain like double chain cubic-spline patterns as shown in Fig. 1, here, the topological neighbors forming the artificial viscoelastic mesh of each robot will contain only the first three available neighbors. With the proposed approach, the control model is able to form self-organized cubic-spline patterns without the requirement of any control points that are basically used in geometry. We also incorporate a potential field-based obstacle/collision avoidance controller to allow robots to avoid the walls surrounding the arena and avoid other collisions between each other (see Section III-B). The control models of our approach are implemented using the open-source ARGoS simulator [13] with a swarm of foot-bot robots (Section IV). The performance assessment of the proposed approach is evaluated first via analyzing the spatial distribution of the swarm and then analyzing the dynamic behavior of the swarm in the absence and presence of obstacles using the averaged mean square displacement (MSD) metric (Section IV-B). Furthermore, we investigated in Section IV-C the effect of the chosen SPH kernel in emerging different *B*-spline-based patterns. Also, in Section IV-D2, we use the quality of the normalized swarm weighted-distance distribution as a modified version of that used in [14] and the averaged mean distance metrics reported in Section IV-D1 to more accurately analyze the performance of the proposed approach. Results within these two metrics with the analysis of the effect of swarm scalability and effect of sensory noise as well as the effect of partially faulty robots are detailed in Sections IV-D3–IV-D5, respectively. Finally, in Section V, we sum up and discuss some of the future perspectives of this study. In what follows and prior to giving the main details of the proposed method, we shall first highlight in Section II some of the related works.

II. Related Works

Flocking aggregation patterns have been widely studied over the past two decades, which has resulted in the development of intense models and theories from several disciplines ranging from biology, statistical physics, control theory to multirobotics [15]. According to early agent-based modeling techniques [16], [17], emerging flocking aggregating patterns can be established and maintained using only a set of repulsive, alignment, and attraction rules between swarm members. Other studies later reported that cohesive like real flocking aggregating patterns do not emerge from a metric distance interaction model as claimed in the early studies, but rather emerge from an interaction model based on the topological distance metric [18], [19].

In this regard, related studies in the literature of multirobotics and swarm robotics, which are close to our work, largely belong to the behavioral approach [7]. In this approach, formation control models leading a group of robots to form self-organizing patterns or to perform a flocking behavior are developed using an implementation of the desired set of behaviors. For example, Lawton *et al.* [20] proposed a behavior-based formation method with an emphasis on maintaining a formation with three mobile robots during the transition from one pattern to another. Xu *et al.* [21] proposed an implementation of a behavior-based design approach applied to control a robotic swarm system while navigating in a large formation in an unknown environment with obstacles. The work of Hauert [6] presented a behavior-based formation control based on a potential field methodology that maintains a minimum distance among members of a large number of agents moving within the desired shaped region. This work was later extended by Haghighi and Cheah [22] via adding a new interactive force to cope with group fragmentation during moving toward the desired formation. Similarly, Hou and Cheah [23] suggested a dynamic formation control for a system of swarm agents using associating multiplicative dynamic potential energy functions with various basic formations, such as circles, triangles, and rectangles. In the same direction and in order to meet the challenge of cooperative control of swarming robots in establishing complex formations, Haghighi and Cheah [24] introduced a formation control concept in which swarms of robots are spat into several groups coordinating to increase the versatility of the entire swarm operating in a complex environment. Also to form regular-based topological formations, such as square and triangular lattices, straight lines, and rings, Zhao *et al.* [25] applied a set of combined artificial force primitives to a system consisting of several distributed robotic nodes. In another work, Zhao *et al.* [26] proposed a self-adaptive solution to the collective motion problematic within a robotics swarm system moving in 3-D space from a source to a destination with a predefined path. While maintaining connectivity and being robust against robot failures and GPS errors, the solution is able to cope with many scenarios in the presence/absence of obstacles with or without leaders in the system.

In other works, potential field approaches are combined with consensus-based control methodologies. The latter is largely inspired by the Reynolds flocking primitives [16] and the Vicsek *et al.* model [17], and are widely studied in the cooperative control of multiagent systems, where agents exchange information with one another for the purpose to reach a consensus of the common value on speeds or headings. For instance, Listmann *et al.* [27] presented a combined potential field and a consensual approach to achieve arbitrary shapes and reactive obstacle behavior with a group of nonholonomic vehicles. Sun *et al.* [28] investigated both the dynamic role assignment and collision avoidance on a formation control with a system of swarm agents using a consensus tracking algorithm. The approach first used a traditional consensus protocol to assign a dynamic role to the team of agents, enforced by applying a modified potential field function to cope with obstacle avoidance. Then, the final formation control is obtained by a modified consensus algorithm containing gradient terms. With a swarm of UAVs, Fu *et al.* [29] designed a fully distributed formation control model in which a combined strategy between the artificial potential field and consensus method is applied so that UAVs can maintain a specific distance while flying in formation with a leader. In another work, Fu *et al.* [30] applied a consensus formation maintenance/reconstruction algorithm imposed by a potential field-based obstacle avoidance control. The method also enabled formations to be transformed by modifying the relative position relationships between individual drones and the virtual leader.

While most of the formation/flocking studies presented above are generally carried out in numerical simulations with simple 2-D/3-D kinematic robot models, a certain number of works have been carried out with real experimental swarm robotics platforms. Most notably, Spears *et al.* [31] introduced the actually well-known *physicomimetics* framework, which provides a distributed control for the self-organization and self-repair of a large number of mobile robots in sensor networks. The framework essentially uses a set of artificial physical laws as the mechanism of interaction between individual robots and it has been effectively applied to construct various regular geometric lattice configurations. In addition, the framework later became the basis for several control models of flocking and pattern formation. For example, Turgut *et al.* [32] studied the self-organized flocking behavior in a swarm of Kobot mobile robots, thanks to the combination of artificial physical forces driven by proximal control and heading alignment. This work was extended later to study the ability of swarm flocking robots to steer toward a specific direction through external guidance of some of its individuals [33]. Likewise, Ferrante *et al.* [10] succeeded in obtaining the flocking behavior with a swarm of mobile robots using a specific self-adaptive communication mechanism, in which the heading direction of the swarm can be controlled to allow the swarm following the desired direction while maintaining cohesion. In another work, Ferrante *et al.* [34] showed that by proposing new motion control, a swarm of mobile robots can flock in a random direction without the need for alignment control or informed robots. With a swarm of small fixed-wing drones, Kownacki and Ołdziej [35] attempted to mimic Reynolds' flocking behavior in reality through only two fundamental-based flocking primitives (cohesion and repulsion) combined with a leadership function.

III. Methodology

Let a swarm of N mobile robots move in a delimited arena and seek to self-organize into cubic-based spline patterns that emerge from local interaction rules. We tackle this problem by following an artificial physics approach, where robots exert artificial forces on each other. In our approach, each robot i of the swarm is subject to the following virtual force law:

$$\hat{f}_i = \hat{g}_i + \hat{o}_i. \tag{1}$$

$\hat{g}_i$ is defined as the cubic-spline formation control law that keeps the robots together and it constitutes the main control in our overall control model. $\hat{o}_i$ is defined as the obstacle/collision avoidance control law that enables the robots to avoid the arena walls and avoid collision between interacting robots.

A. Cubic-Spline Formation Control

To achieve this controller, we assume that the robots are capable to communicate with each other within a communication range Cr. The set of the ith robot's neighbors at time step t within Cr is denoted by $\mathcal{N}_i(t)$, and ϕ_{ij} and l_{ij} are, respectively, the relative bearing and range of the jth neighbor of the robot, i. The idea behind this control is that an artificial viscoelastic mesh (AVM) [14], [36] is built among the robot's three available nearest neighbors ($\mathcal{K}_i^3, \mathcal{K}_i^3 \in \mathcal{N}_i$) having the smallest weighted distances. The overall chart illustrating the steps of constructing each robot's AVM and which constitute the main control steps of the cubic-spline aggregation controller is shown in Fig. 2. The steps, which are executed by each robot, i of the swarm, are detailed as follows.

1) *Density Estimation:* The first step in our control is to estimate the robot's density ρ_i^3 within the swarm using

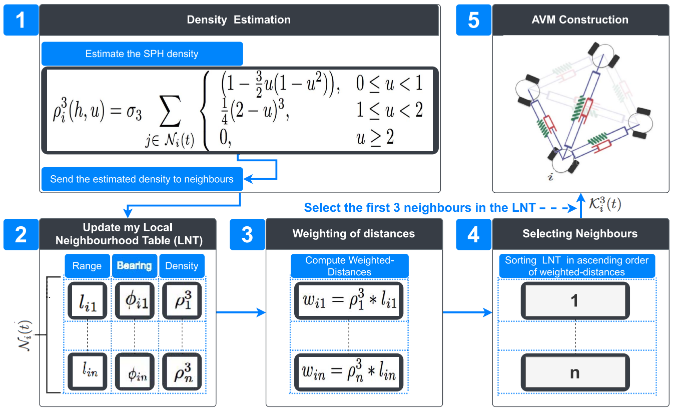

1
Density Estimation
Estimate the SPH density
$\rho_i^3(h,u) = \sigma_3 \sum_{j\in \mathcal{N}_i(t)} \begin{cases} \left(1-\frac{3}{2}u(1-u^2)\right), & 0 \le u < 1 \\ \frac{1}{4}(2-u)^3, & 1 \le u < 2 \\ 0, & u \ge 2 \end{cases}$
Send the estimated density to neighbours
5
AVM Construction
Select the first 3 neighbours in the LNT
$\mathcal{K}_i^3(t)$
2
Update my Local Neighbourhood Table (LNT)
Range
Bearing
Density
l_{i1}
ϕ_{i1}
ρ_1^3
l_{in}
ϕ_{in}
ρ_n^3
$\mathcal{N}_i(t)$
3
Weighting of distances
Compute Weighted-Distances
$w_{i1} = \rho_1^3 * l_{i1}$
$w_{in} = \rho_n^3 * l_{in}$
4
Selecting Neighbours
Sorting LNT in ascending order of weighted-distances
1
n

Fig. 2. Overall chart of the cubic-spline formation control model.

an SPH density estimation technique [37]. Note that SPH is a Lagrangian-based method widely used in solving the hydrodynamics equations of fluid dynamics. It uses generally a set of kernel interpolation functions to approximate physical quantities moving with particles [11]. While several kernel functions can be applied in SPH, we refer here to the default M_4 cubic-spline ones that have being used over the last 30 years [37] as follows:

$$\rho_i^3(h,u) = \sigma_3 \sum_{j\in \mathcal{N}_i(t)} \begin{cases} \left(1-\frac{3}{2}u\left(1-u^2\right)\right), & 0 \le u < 1 \\ \frac{1}{4}(2-u)^3, & 1 \le u < 2 \\ 0, & u \ge 2 \end{cases} \tag{2}$$

where $\sigma_3 = 10/(7\pi h^2)$ is a constant that normalizes the cubic-spline functions of parameter u ($u = 2[l_{ij}/C_r]$), and h is the kernel smoothing length. Note that the idea here is to view robots as particles currying physical quantities. Here, the M_4 cubic-spline kernel functions are used to interpolate the robots' densities as physical quantities moving with the robots. This information that is diffused over the robot's neighbors will then used as key information in emerging cubic-spline patterns.

2) *Local Neighborhood Table (LNT) Updating:* The estimated density computed in the first step and which is diffused (communicated) to the robot's neighbors is used to construct and update the robot's LNT. The LNT contains neighborhood information in a matrix form of size $|\mathcal{N}_i(t)|$x3. Each line of the matrix holds information $(l_{ij}, \phi_{ij}, \rho_j^3)$ about each jth neighbor of the set $\mathcal{N}_i(t)$.
3) *Weighting of Distances:* Now, using the information in the LNT the corresponding distance of each robot's jth neighbor can be weighted simply by: $w_{ij} = \rho_j^3 * l_{ij}$. Note that by weighting the distances toward neighbors, the impact of a neighbor robot j holding a heavy density and located far away from the robot i could be greater than the other holding a weak density and sitting close to the robot i.
4) *Selecting Neighbors:* In this step, a robot i have to select only three neighbors if available among the set of all its sensed neighbors, $\mathcal{N}_i(t)$. The selection is based on the weighted-distances computed in the previous step by sorting the LNT according to those having the smallest w_{ij}. The first three available ones among the sorted LNT are then selected to be a part of the set $\mathcal{K}_i^3(t)$.
5) *AVM Construction:* Now, the AVM of each robot i is composed of three viscoelastic links, each represented by a springer and a damper connected in parallel. The resulting force driven by the AVM is given as follows:

$$\hat{g}_i = \sum_{j\in\mathcal{K}_i^3(t)} \left(\alpha\left(l_{ij}-l_0\right)\hat{l}_{ij} + \beta\left(v_i - v_j\right)\right) \tag{3}$$

where α and β are, respectively, the spring and damping coefficients of each viscoelastic link and l_0 is the equilibrium length of the spring serving as the desired distance between the robots of the mesh. $\hat{l}_{ij}$ represents the direction of the virtual spring force vector and v_i and v_j are, respectively, the speeds of the interacting robots. Note the fact that each robot's AVM is formed by only three virtual viscoelastic links (see the figure inside the box in step 5 of Fig. 2), the resulting basic geometric formation driven by the $\hat{g}_i$ will be then a rhombus shape. As a consequence, the global entire swarm mesh will lead then to emerge double-chains such as cubic-spline-based patterns, since the inter-robot distances are weighted using the M_4 cubic-spline-based SPH kernel.

B. Obstacle and Collision Avoidance Control

To guarantee that each robot avoid the walls surrounding the arena as well as to avoid collision between its remaining neighbors that are not belonging to the $\mathcal{K}_i^3(t)$' set, a repulsive potential field is generated around the closest wall as well as the closest neighbor to the robot. The potential field will push away every robot close to the influenced zone of the field. This control is achieved as follows [38], [39]:

$$\hat{o}_i = \begin{cases} \epsilon\left(\frac{1}{\ell} - \frac{1}{\ell_0}\right)\left(\frac{1}{\ell^2}\right), & \ell \le \ell_0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where ϵ is a gain constant, ℓ is the distance to the closest obstacle or neighbor robot, and ℓ_0 represents the obstacle influence threshold.

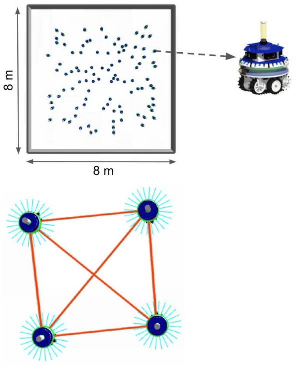
C. Robot's Motion Control

The robots we used in this study are two wheels differential drive mobile robots. The motion of such type of robots is driven by the forward speed of the robot's left and right wheels as follows:

$$\begin{bmatrix} v_{l_i} \\ v_{r_i} \end{bmatrix} = \begin{bmatrix} 1 & \frac{d}{2} \\ 1 & \frac{-d}{2} \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{5}$$

d represents the distance between the robot's wheels. v_i and ω_i represent the forward and the angular robot's velocities, respectively, which can be computed as follows: now, the AVM force vector, $\hat{f}_i$ can be used to compute these velocities as follows:

$$\omega_i = \xi \measuredangle \hat{f}_i, v_i = \frac{v_{max}}{\sqrt{\mid \omega_i \mid +1}} \tag{6}$$



Parameter	Description	Value(s)
N	Number of the foot-bots	{50,75}
C_r	Communication range	1 m
α	Spring gain constant	1.9
l_0	Equilibrium length of the spring	0.15 m
β	Damper gain constant	1.25
ϵ	Obstacle gain constant	0.01
ℓ_0	Obstacle influence threshold	0.01 m
ξ	Angular scalar constant	0.95
d	Inter-wheels distance	0.14 m
$v_{\mathfrak{max}}$	Maximum forward speed	0.08 m/s
h	The smoothing length	0.5 m
T	Total simulation time steps	3000
t	Time step duration	0,1 sec

Fig. 3. (Top left) View of the arena settings with a zoom-in of a simulated foot-bot robot (top left), (bottom left) view of four simulated foot-bots interacting with each other via the AVM model: blue sky colors represent the 24 IR proximity sensors and the red lines represent the communication RAB device. (Right) Different simulation constants and parameters used in our simulations.

where ξ is an angular scalar constant and $v_{\mathfrak{max}}$ is the robot's maximum allowed forward speed.

IV. Simulation Results, Analysis, and Discussion

A. Simulation Settings

For the validation of our approach, we implemented the different subcontrollers cited in Section III using the ARGoS multirobot simulator [13]. We particularly conducted experiments with two different swarm sizes using a simulated swarm of foot-bot robots placed arbitrarily inside a square arena of 8×8 m^2 of surface (see the top left-hand side of Fig. 3). A foot-bot is a two-wheeled differential drive mobile robot that has been widely used as a swarm robotics platform testbed. The robot can send/receive messages to/from nearby robots through its range and bearing (RAB) communication device. It can also detect objects thanks to its 24 IR proximity sensors distributed equitably around its body (see the bottom left-hand side of Fig. 3). The RAB device is used specifically to achieve the cubic-spline aggregation subcontroller, whereas the 24 IR proximity sensors are used to achieve the obstacle/collision avoidance subcontroller. Note that in our experiments, the RAB measures of the RAB device are noised using a Gaussian noise model of the form $\mathcal{N}(0, 0.01)$ and the probability of communication failure is fixed at 3%. The parameters and constants related to all subcontrollers are summarized in the table displayed in Fig. 3.

B. ARGoS-Based Simulation Results

We conducted a number of ARGoS-based experimental simulations following the settings of Section IV-A. Fig. 4 highlights a few snapshots during an experimental simulation of a swarm robots system performing the proposed controller in the absence [Fig. 4(a)] and presence [Fig. 4(b)] of obstacles inside the arena. The screenshots are taken at different time steps $t = \{0, 500, 1500, 3000\}$ with a swarm size $N = \{50, 75\}$. In both scenarios, it is seen the achievement of self-organized cubic-spline patterns even in the presence of obstacles. The swarm starts to be in these self-organized cubic-spline patterns at time step $t = 500$, and that the patterns emerge clearly at the final stage of the simulation. Moreover, the with obstacles case study shown in Fig. 4(b), in which four obstacles of size ($0.25 \times 0.5 \times 0.25$ m) are randomly placed in the arena, indicates how the swarm can smoothly avoid the obstacles while maintaining the patterns. This allows the approach to be applied in attracting, for example, a large swarm of robots from one region to another while avoiding collisions and maintaining connectivity between components of the swarm. This is explained by the fact the robots are mainly governed by the cubic-spline aggregation controller reported in Section III-A. Note also the fact that within this controller, the robots are attracted to only the 3 available neighbors having the smallest weighted distances. This leads to the emergence of double-chain-like cubic-spline formations as it can be seen clearly from the snapshots in Fig. 4.

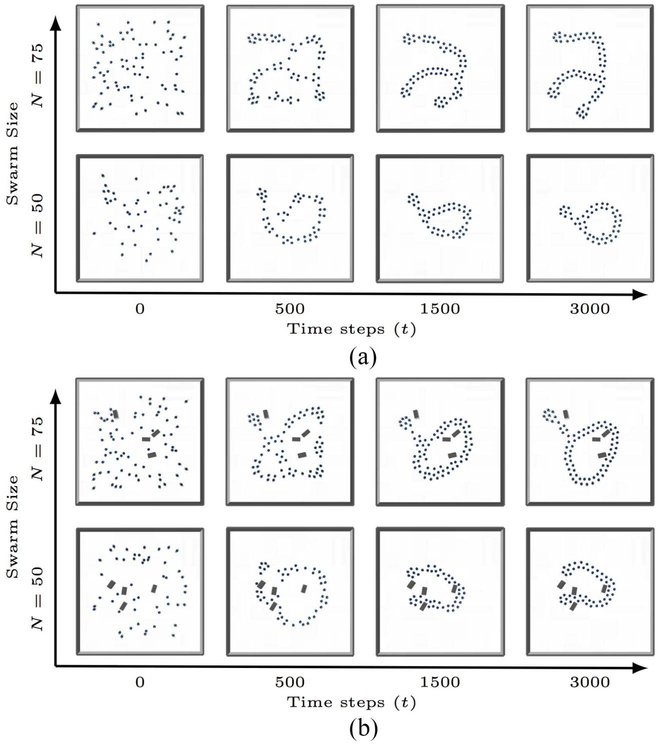
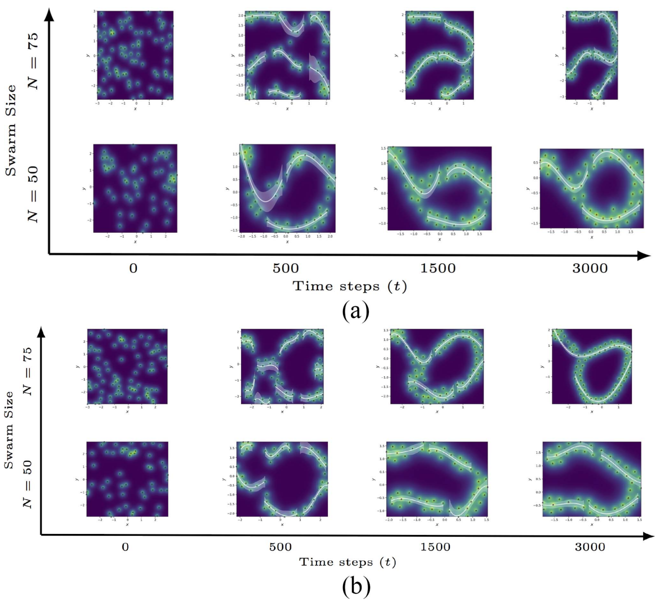


Fig. 4. ARGoS snapshots at time steps $t = \{0, 500, 1500, 3000\}$ showing emerging cubic-spline patterns with a swarm size $N = \{50, 75\}$ in (a) absence and (b) presence of obstacles.

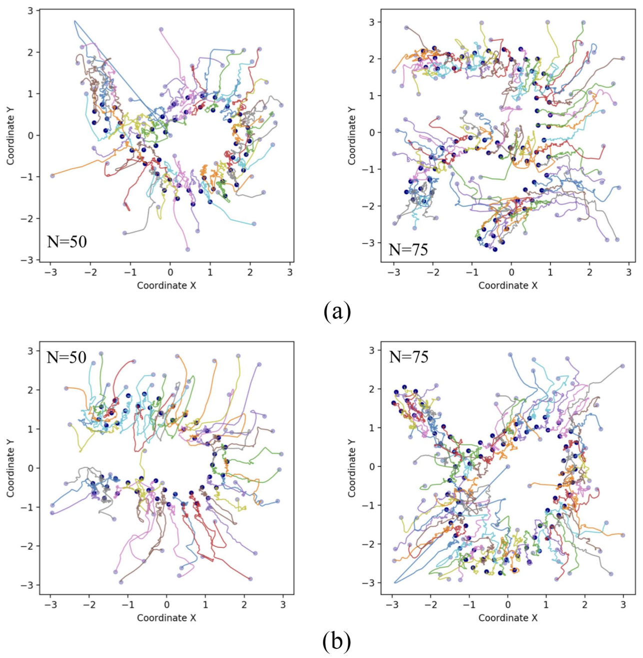
Furthermore, to analyze the cubic-spline patterns emerging from the above results, we analyzed at time steps $t = \{500, 1500, 3000\}$, the spatial distribution of the swarm by plotting the density map of the robots positions [see Fig. 5(a) and (b)] with highlighting the fit of the corresponding spline curves (white curve lines with transparent white bands representing $\pm$*std*-interval). Note that the grids for these density maps match only the minimum bounding zones of the simulated area that encompass the entire swarm and that the spline curves fittings are of degree 3. Note also that these spline curves are generated by first clustering the robots' positions using the K-means clustering algorithm. Then, we interpolate the positions of the robots belonging to each cluster by the best polynomial curve fit of degree n. We found that the best



(a)

(b)

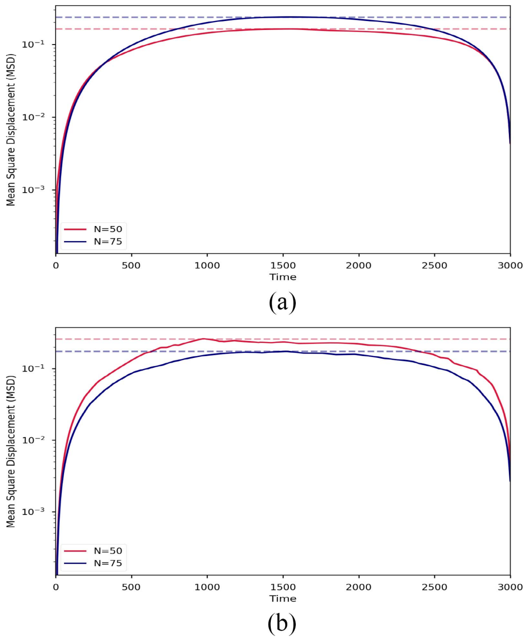
Fig. 5. Swarm robots density map with spline curves fitting of degree 3 (white line) and ±*std*-interval (white band) at time steps $t = \{0, 500, 1500, 3000\}$ with a swarm size $N = \{50, 75\}$ in (a) absence and (b) presence of obstacles.



(a)

(b)

Fig. 6. Robots trajectories for swarm sizes $N = 50$ (left) and $N = 75$ (right) in (a) absence and (b) presence of obstacles scenarios.



(a)

(b)

Fig. 7. Swarm robots average MSDs in two scenarios. (a) Without obstacles. (b) With obstacles.

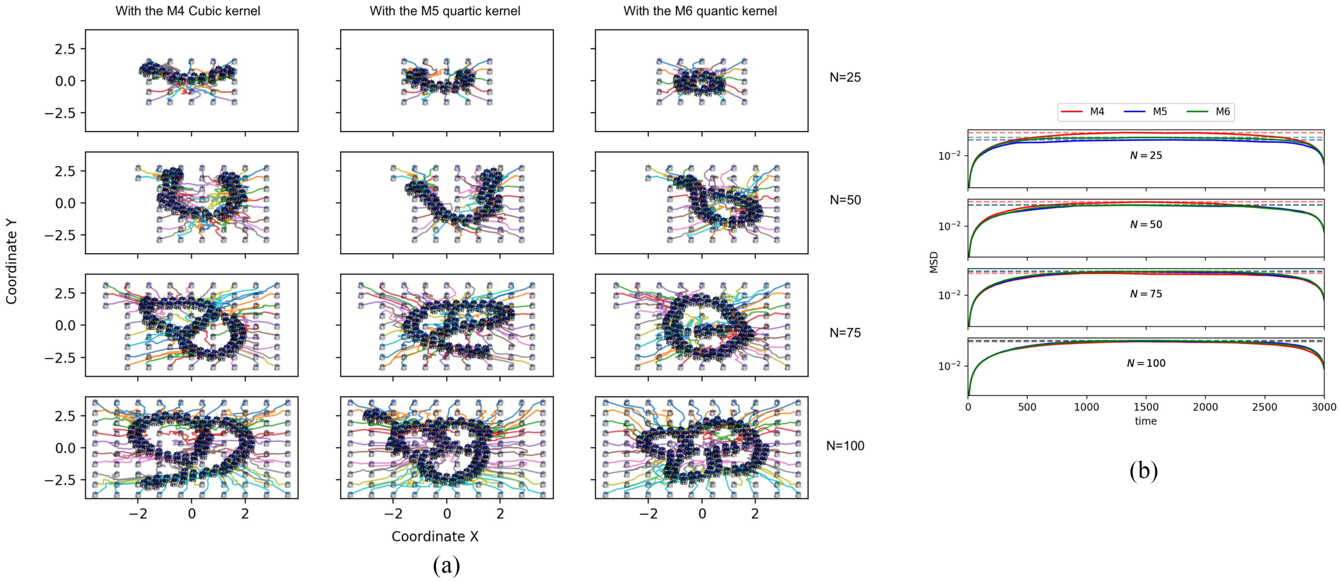
one with all the tested $n \in [2, \ldots, 8]$ values was the ones of degree 3. This is as expected since the kernel functions used in our main SPH-based aggregation control model are of type cubic spline.

Moreover, we analyzed the dynamic behavior of the entire swarm by analyzing the movement of its individual robots. Fig. 6(a) and (b) shows respectively the trajectories followed by the robots in the absence and presence of obstacles for both swarm case studies ($N = \{50, 75\}$), starting from the initial positions at time step $t = 0$ to the final position at time step $t = 3000$. Analyzing the dynamic behavior of the entire swarm with a log scale of the averaged MSD depicted in Fig. 7(a) and (b) shows that the swarm in both scenarios with both sizes ($N = \{50, 75\}$) moves fast initially before slowing down to reach the maximum displacement at approximately $t = 1500$ (see dashed horizontal lines). At this point, we observed that self-organized cubic-spline patterns begin clearly to emerge. The swarm later starts first reducing slowly its movement to stabilize the patterns than accelerating quickly to decrease its displacement until the end of the simulation. However, in the with-obstacles case study illustrated in Fig. 7(b), it is evident that the swarm MSD suffers perturbations during a few periods of simulation time induced by the trajectories of the robots tending to deviate from the encountered obstacles.

C. Effect of SPH Kernels in Emerging Self-Organized Spline Patterns

In this section, we are more interested in studying how the kernels used in the SPH estimation technique can emerge different self-organized spline patterns. As noted in the cubic-spline formation control section, there exist other kernel functions in the SPH literature that differs from the M_4 kernel and which might be applied in our case. To this end, we investigated the effect of two other kernels based on the Schoenberg *B*-spline functions [11] that are M_5 (quartic) and M_6 (quantic) kernels and we compared the obtained results with that of the M_4 (cubic) kernel (see Table I for further description of the M_5 and M_6 SPH kernels).

Following the same settings of Section IV-A, 25 experimental simulations were conducted using the ARGoS simulator with a swarm of different sizes ($N = \{25, 50, 75, 100\}$) initially distributed within an obstacle-free region in 5 × 5, 8 × 8, 9 × 9, and 10 × 10 grid layouts, respectively. Fig. 8(a) shows exactly the robots' initial positions, their trajectories, and their



(a)

(b)

Fig. 8. (a) Self-organized *B*-Spline formations obtained with a swarm of different sizes, initially distributed in a grid, performing our approach while using different chosen kernels in the SPH interpolation technique. (b) Corresponding swarm average MSDs.

TABLE I
M_5 (QUARTIC) AND M_6 (QUINTIC) SCHOENBERG *B*-SPLINE KERNEL FUNCTIONS [11] USED IN OUR STUDY

The Kernel Name	The Kernel Functions	The Normalized Constant (σ_3)
M_5	$\rho_i^3(h,u) = \sigma_3 \sum_{j\in \mathcal{N}_i(t)} \begin{cases} (\frac{5}{2}-u)^4 - 5(\frac{3}{2}-u)^4 + 10(\frac{1}{2}-u)^4, & 0 \leq u < \frac{1}{2} \\ (\frac{5}{2}-u)^4 - 5(\frac{3}{2}-u)^4, & \frac{1}{2} \leq u < \frac{3}{2} \\ (\frac{5}{2}-u)^4, & \frac{3}{2} \leq u < \frac{5}{2} \\ 0, & u \geq \frac{5}{2} \end{cases}$	$\frac{96}{(1199\pi h^{2.5})}$
M_6	$\rho_i^3(h,u) = \sigma_3 \sum_{j\in \mathcal{N}_i(t)} \begin{cases} (3-u)^5 - 6(2-u)^5 + 15(1-u)^5, & 0 \leq u < 1 \\ (3-u)^5 - 6(2-u)^5, & 1 \leq u < 2 \\ (3-u)^5, & 2 \leq u < 3 \\ 0, & u \geq 3 \end{cases}$	$\frac{7}{(478\pi h^3)}$

final positions obtained from averaging the results of the conducted experiments while performing the formation controller with M_4, M_5, and M_6 kernels. It can be clearly seen from the figure that when choosing different kernels to estimate the density of the robots in the swarm, various *B*-Spline-based patterns could emerge accordingly while starting from the same robots' initial positions. This is as expected since the kernels used to estimate the density of the swarms are all based on a set of *B*-Spline interpolation functions.

Furthermore, when analyzing the swarm dynamic behavior of the above-conducted experiments with MSD [see Fig. 8(b)], we found that when applying M_5 and M_6 kernels, the global swarm dynamic is almost equal. Also, as the swarm size increases, the displacement of a swarm of the same size follows nearly the same dynamic behavior when applying M_4, M_5, and M_6 kernels. However, when the size of the swarm is small ($N = 25$), the displacement of the swarm with the M_4 kernel is slightly longer than the M_5 and M_6 case studies. As expected, the swarm for all sizes case studies with any chosen kernel attains and maintains the maximum displacement at nearly $t = 1500$ and this after being exponentially accelerating its displacement from the beginning. At $t = 1500$, self-organized based *B*-spline patterns clearly began to emerge from the swarm interactions. From that point, the swarm starts exponentially decreasing its displacement with slow decay till the end of the simulation, meaning that the swarm is almost maintaining the formation of the emerging self-organized *B*-spline-based patterns.

D. Analysis Results and Discussion

In this section, we are interested in assessing the performance of the proposed method via evaluating the dynamic behavior of the swarm throw the following performance metrics.

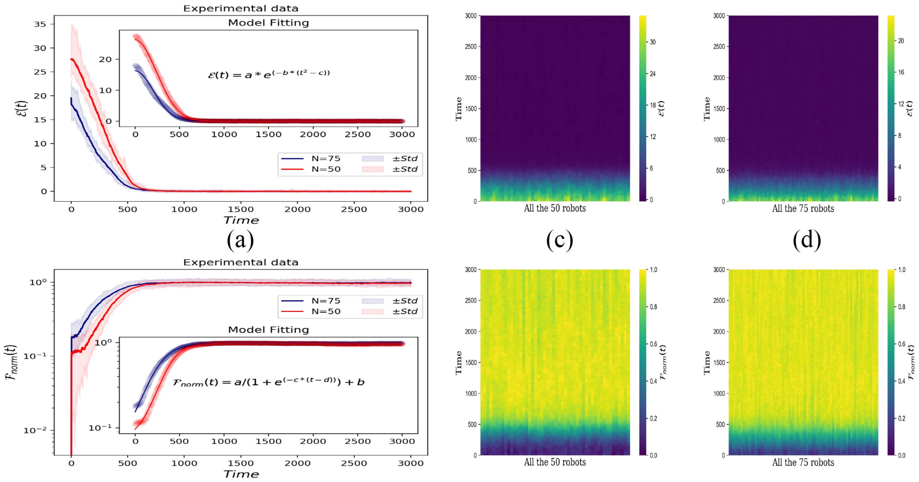
1) Performance Metrics Formulation:

1) *Averaged Mean Distance Error:* We rely on this metric to measure the quality of maintaining a desired distance among interacting robots as they are moving together. It can be defined as the inter-robots distance error averaged over the different robots and neighbors as follows:

$$\mathcal{E}(t) = \frac{1}{KN} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \left(l_{ij}(t) - l_0 \right) \right). \tag{7}$$

Here, $k <= 3$ represents the number of the robot's neighbors forming the AVM at time t. The most $\mathcal{E}(t)$ is close to 0, the most the desired distance is maintained on the swarm.

2) *Normalized Weighted-Distance Distribution Quality:* In order to assess the distribution quality of the overall



(a) (c) (d)

(b) (e) (f)

Fig. 9. Time convergence of (a) $\mathcal{E}(t)$ and (b) $\mathcal{F}_{\text{norm}}(t)$ with swarm size $N = \{50, 75\}$. Curves represent the mean values and its corresponding $\pm Std$ obtained from 100 simulations with different initial configurations of robots. (c) and (d) Heatmaps showing each robot's $\mathcal{E}(t)$ value for $N = 50$ and $N = 75$, respectively. (e) and (f) Heatmaps showing each robot's $\mathcal{F}_{\text{norm}}(t)$ value for $N = 50$ and $N = 75$, respectively. In all the heatmaps, the values of the metrics are averaged over 100 simulations.

swarm weighted-distances, we adopted a modified version of the distance-weighted distribution quality used in [14] and [36] as follows. First, we calculate the weighted distances $\mathcal{A}_i(t)$ of each robot as the average of the different robot's neighbors forming the AVM as follows:

$$\mathcal{A}_i(t) = \frac{1}{K}\left(\sum_{j=1}^{K} w_{ij}(t)\right). \tag{8}$$

Second, the robot's weighted-distance quality metric $\mathcal{F}_i(t)$ is obtained by the following equation as in [14] and [36]:

$$\mathcal{F}_i(t) = 1 - \frac{1}{\sqrt{\mathcal{A}_i(t)+1}}. \tag{9}$$

Then, the swarm normalized modified weighted-distance distribution quality $\mathcal{F}_{\text{norm}}(t)$ averaged over all the robot swarm size is calculated as follows:

$$\mathcal{F}_{\text{norm}}(t) = \sum_{i=1}^{N} \frac{\mathcal{F}_i(t)}{\max(\mathcal{F}_i(t))}. \tag{10}$$

Here, the normalized metric serves to better compare the swarm performance whatever the swarm size is. Note that the possible values will be between the range [0, 1] and that a higher value signifies a better swarm performance.

3) *Time to Be in Formation:* This measure represents the time that the swarm takes to be in a formation. To calculate this metric, we first define the function: $IIF(t)$ that determines if the swarm is in formation as follows [40]:

$$IIF(t) = \begin{cases} 1, & \mathcal{E}(t) \le e \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

so that e is a certain fixed-error threshold. Then, the time to be in formation TIF can be defined simply based on the time taken to reach $\mathcal{E}(t) \le e$ [40].

2) *Performance Results Discussion:* Following the same settings in Section IV-A, 100 trials with different initial robots configuration per swarm size experimental simulation are conducted. ARGoS-based data are collected at each time step of the simulation and the corresponding results of $\mathcal{E}(t)$ and $\mathcal{F}_{\text{norm}}(t)$ metrics are depicted in Fig. 9. In both plots, the curves in solid lines represent the median values of the 100 trials and the transparent colored bands represent the according $\pm Std$ values.

From Fig. 9(a), it can be clearly seen that $\mathcal{E}(t)$ follows nearly a fast exponential decay model, where the swarm starts exponentially decreasing the mean distance error until it enters the stationary phase. Our results suggest that during approximately the first $t \in [480, 580]$ time steps for swarm size $N = 75$ and $t \in [580, 680]$ for swarm size $N = 50$, the rate of decay is greatly decreased and that after this short period of time, the desired distance between the robots is maintained during the remaining simulation time steps. In contrast from Fig. 9(b), it is clearly observed that the log-scaled results of $\mathcal{F}_{\text{norm}}(t)$ follow approximately an exponential growth model. Results indicate that the swarm robots system tends successfully to converge into a stable maximum value (approximately 1) for both swarm size case studies. Note that the swarms achieve the stationary phase within the same time convergence of $\mathcal{E}(t)$. This means that the final self-organized cubic-spline patterns emerge while all the robots have nearly the same SPH density with maintaining the same desired distance among themselves. The numeric solutions for the best mathematical models fitting our experimental data, and which are shown in the embedded plots of Fig. 9(a) and (b), are presented in the tables of Fig. 10.

Furthermore, for better visualization of the swarm performance under the above settings, the performance results of each robot in the swarm—for both swarm sizes case studies—are plotted using heatmaps [see Fig. 9(c)–(f)]. For all the heatmaps, the x axis represents all the robots of the swarm and the y axis represents the time simulation

Model Fitting for $\mathcal{E}(t)$	
$\mathcal{E}(t) = a.e^{-b.(t^2-c)}$	
For Swarm Size $N = 50$	For Swarm Size $N = 75$
$a = 1.58207000.10^{+1}$	$a = 2.61149183.10^{+1}$
$b = 1.18849863.10^{-5}$	$b = 8.68147578.10^{-6}$
$c = 2.46531350.10^{+3}$	$c = 1.25412696.10^{+3}$

Model Fitting for $\mathcal{F}_{norm}(t)$	
$\mathcal{F}_{norm}(t) = a/(1 + e^{(-c.(t-d))} + b$	
For Swarm Size $N = 50$	For Swarm Size $N = 75$
$a = -9.49923224.10^{-1}$	$a = -9.01531360.10^{-1}$
$b = 9.85409923.10^{-1}$	$b = 9.74965528.10^{-1}$
$c = -7.84187583.10^{-3}$	$c = -9.69380931.10^{+3}$
$d = 2.50055448.10^{+2}$	$c = 3.76975699.10^{+2}$

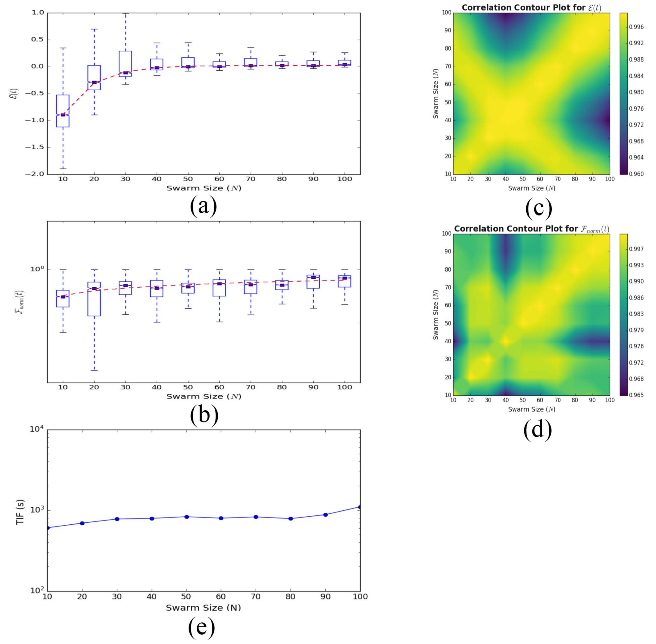
Fig. 10. Numerical solutions for the best models fitting our experimental data.

$t \in [0..3000]$. The colors in the heatmaps represent the corresponding performance metric values of each robot. As can be seen in Fig. 9(c) and (d), nearly a sixth of each $\mathcal{E}(t)$' heatmap is having high values (see yellow and dark green colors at the bottom of the corresponding heatmap). Whereas, the remaining part of the heatmaps are having very low values that approximate 0 (dark navy colors). In contrast, the heatmaps in Fig. 9(e) and (f) presenting the results of $\mathcal{F}_{\text{norm}}(t)$ for both swarm sizes case studies report that almost a sixth of the heatmaps are having low values (dark navy and dark green colors at the bottom) and that the rest parts of the heatmaps are having nearly very hight values (see yellow and light green colors at the top). The fact that nearly a sixth of the parts of the heatmaps are having almost the same color, which means that at roughly a sixth of the total time simulation ($t \approx 500$), all the robots start to converge to a steady metric value. This confirms our results obtained for the entire swarm which reported that the final cubic-spline-based patterns emerge while all the robots are maintaining the same desired distance among themselves while having nearly the same SPH density.

Moreover, the obtained results within the $\mathcal{E}(t)$ and $\mathcal{F}_{\text{norm}}(t)$ metrics can be confirmed when computing the TIF(t) metric for the same case scenarios. The corresponding TIF results with $e = 0.5$ cm of threshold confirm that the swarm takes about 570 and 642 time steps to be in formation for $N = 50$ and $N = 75$, respectively. This suggests that the swarm of size $N = 50$ and $N = 75$ maintained in formation for almost 80% and 82% of the total simulation time, respectively.

3) Robustness to Swarm Scalability: In this section, we are interested in investigating the scalability of our approach with respect to increasing the size of the swarm. This time, we take into account different swarm sizes ($N = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$) and we conducted 100 runs per swarm size with different initial robot configurations. The other settings of each simulation were kept the same as in the previous sections. In each run, we collected data from the start to the end of the simulation, meaning that both the exponentially and the stationary phases of the time convergence of the metrics are taken into consideration in this study.

Fig. 11 presents box plots of the time evolutions of $\mathcal{E}(t)$ [Fig. 11(a)] and $\mathcal{F}_{\text{norm}}(t)$ [Fig. 11(b)] with respect to the swarm size N. Each box plot is a representation of the metric values collected from 100 runs. A least square regression fit of the form: $-a.B^x + c$ (dashed red lines) is performed to the mean metrics values that are represented by blue squares inside the box plots. The numeric solution for $\mathcal{E}(t)$ yields $a \approx 2.57$, $B \approx 0.90$, and $c \approx 0.02$ while for $\mathcal{F}_{\text{norm}}(t)$, it results in $a \approx 0.06$, $B \approx 0.99$, and $c \approx 1.01$. For the obtained solutions, it seems that both scaled exponentially in the time taken to emerge self-organized cubic-spline-based patterns. Note that the scale rate growth solution for $\mathcal{E}(t)$ is clearly observable in comparison to that of $\mathcal{F}_{\text{norm}}(t)$, and that while the swarm size is small (less than 20), the performance of the swarm is observed decreasing. This could happen in situations where a number of the robots of the swarm are initially largely dispersed in the arena far away from each other, leading to not detecting neighboring robots. Therefore, these robots will move straight line until they detect neighbors and start forming the AVM or an obstacle (an arena wall) and immediately deviate from it and continue moving forward in seeking to encounter neighboring robots.



(a)

(b)

(e)

(c)

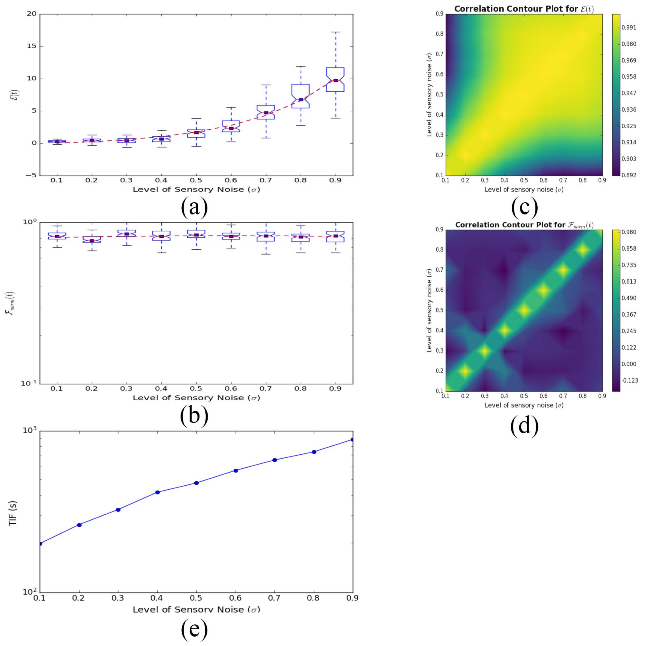
(d)

Fig. 11. Effect of the swarm size N on the performance of the swarm: box plots with least square regression fit for (a) $\mathcal{E}(t)$ and (b) $\mathcal{F}_{\text{norm}}(t)$, correlation contour plot for (c) $\mathcal{E}(t)$, and (d) $\mathcal{F}_{\text{norm}}(t)$, and (e) time to be in formation.

Also, it can be clearly seen, particularly from Fig. 11(a), that the swarms almost perform equally while the swarm size begins to increase from $N = 40$ to $N = 100$. These results are confirmed while highlighting visual illustrations of the correlation between the ten case studies of swarm size N plotted in the form of correlation contour plots in Fig. 11(c) and (d).

One more interesting observation is that despite the time to be in the formation [see Fig. 11(e)] increases slightly with the number of robots in the swarm, which is clearly a result of the fact that the larger the swarm, the more interactions are required. However, the increase rate is remarkably small due to the fact that the inter-robot interactions in our proposed approach are governed only by the three available neighbors regardless of swarm size.

4) Robustness to Sensory Noise: We study in this section the effect of sensory noise on the performance of the proposed solution. We aim particularly to investigate how the cubic-spline aggregation control model will behave when the readings of the RAB sensors of the foot-bots are corrupted by noise. For this purpose, we applied different Gaussian noise models in the form $\mathcal{N}(0, \sigma)$ to the readings of the RAB sensors



(a) (c) (b) (d) (e)

Fig. 12. Effect of the RAB sensory noise level σ on the performance of a swarm composed of 75 foot-bots with $\sigma \in \{0.1, ..., 0.9\}$: Box plots with least square regression fit for (a) $\mathcal{E}(t)$ and (b) $\mathcal{F}_{\text{norm}}(t)$, correlation contour plot for (c) $\mathcal{E}(t)$ and (d) $\mathcal{F}_{\text{norm}}(t)$, and (e) time to be in formation.

while performing the proposed control model with a swarm size $N = 75$. The σ values used in our study are in the range of $[0.1, 0.9]$ with an increased step of 0.1. The other setting parameters described in the table of Fig. 3 are kept the same. Within this setting, we performed 100 simulation runs per σ value and we collected ARGoS-based data per simulation run from the start till the end of the simulation. The red dashed lines in the figure represent the least square regression fitting curves of the form: $a.B^x + c$ for both figures. The numerical solution for fitting the $\mathcal{E}(t)$ median values yields $a \approx 0.30$, $B \approx 1.48$, and $c \approx -0.43$. Whereas, the solution leading to fit the $\mathcal{F}_{\text{norm}}(t)$ median values gives $a \approx 86.08$, $B \approx 1$, and $c \approx -85.27$.

Results of this study are depicted in Fig. 12. The results are shown in box plots representing the values of $\mathcal{E}(t)$ [Fig. 12(a)] and $\mathcal{F}_{\text{norm}}(t)$ [Fig. 12(b)] obtained from performing the 100 simulation runs per σ value. While Fig. 12(a) makes it clear that the desired inter-robot distance is being affected exponentially by the level of noise present in the readings of the robots RAB device, and this is as expected since the $\mathcal{E}(t)$ metric is strongly related to the readings of this sensor. However, the normalized weighted-distance distribution quality ($\mathcal{F}_{\text{norm}}(t)$) in Fig. 12(b) is almost unaffected by this fact (an almost fixed red line due to $B \approx 1$). Our results of $\mathcal{E}(t)$ also suggest that the slope of the exponential growth nearly begins when the level of noise σ is approximately 0.6, meaning that the swarm with our approach still performs well and still tolerant to noises with up to 0.5.

These results are confirmed also in the visual correlation contour plots shown in Fig. 12(c) and (d), respectively. The plots visualize for each performance metric the correlation between the ten case studies of the noise level σ presented in the robots' RAB devices. The correlation contour plot in Fig. 12(c) makes it clear how much the level of noise presented in the RAB device can affect the performance of the swarm in maintaining a desired inter-robots distance. The more we increase σ, the higher the correlation is (yellow-colored regions). In contrast, the correlation contour plot in Fig. 12(d) reporting the results of $\mathcal{F}_{\text{norm}}(t)$ shows the robustness of our approach to sensory noise (very low correlation among all the case studies of σ values).

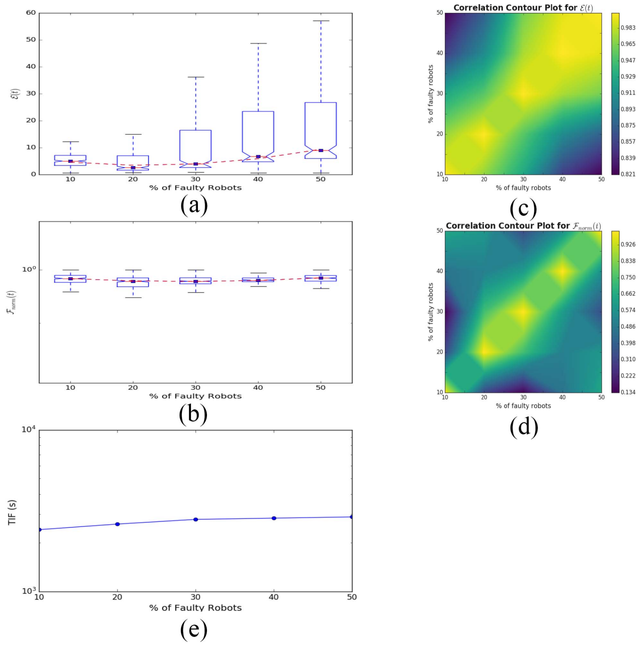
Furthermore and as expected, the time that the swarm takes to be in the formation from this study [see Fig. 12(e)] clearly confirms that the higher the level of noise is contaminated in the robots' RaB sensors, the longer it would take for the swarm to be in the formation.

5) Robustness to Faulty Robots: In real application scenarios, it may cause some of the individual robot components to fail to perform due to external or internal faults. A robust control model should be tolerant to a certain level to such occurring faults. Here, we are dealing with a certain type of faults called partial faults, which has been specifically reported to have more impact in perturbing the overall performance of robotics swarm systems while compared to complete faults type [41]. Note that a partial fault type occurs when a robot's component does not stop completely to perform, but one of its subcomponents is damaged or does not function properly.

To assess how our formation control model is tolerant to such types of faults, we conducted experimental simulation studies with two partially faults modes: 1) a partial motor failure and 2) a partial RAB device failure. In the partial motor failure type, we assume that one robot's motor—the right or the left one—fails to function properly during certain periods of the total time running. Similarly, in the second failure type, we assume that half of the robot's RAB IR sensors are not functioning well, which is strongly reflected in the fact that it does not allow the robot to detect neighbors that are in a bearing angle of range $[0, \pm\pi]$ depending on the damaged half.

To this end, we only consider that the left motor of the defective robots is misfunctioning in the study of partial motor failure case study and that the second half of the robot's IR RAB sensors are malfunctioning when conducting the RAB partial failure case study. Following the same parameters in Section IV-A, we perform 50 trials per experimental study of partial failure type with a swarm of 75 robots and each with the hypothesis that $n \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ of the robots are partially affected. In each experimental case study, the injection of the appropriate partial fault type is programmed in real time at two distinct periods ($t \in [100, 1200]$ and $t \in [1500, 2200]$) of the total ARGoS time simulation ($t = 3000$). Note that in all the case studies, the robots chosen to be faulty are identified randomly and that these robots will only perform properly 40% of the total running time simulation.

Results of the partial motor fault type and the partial RAB failure-type case studies are depicted in Figs. 13 and 14, respectively. The obtained results suggest that the swarm averaged mean distance error is slightly affected by the number of robots affected by a partial motor failure [see Fig. 13(a)].



(a)

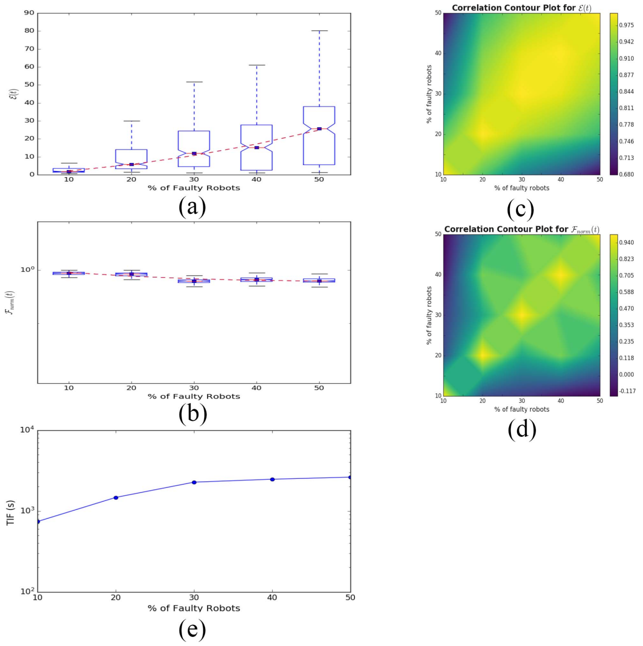
(c)

(b)

(d)

(e)

Fig. 13. Effect of $n \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ faulty robots affected by a partial motor failure type on the performance of a swarm composed of 75 foot-bots: box plots with least square regression fit for (a) $\mathcal{E}(t)$ and (b) $\mathcal{F}_{\text{norm}}(t)$, correlation contour plot for (c) $\mathcal{E}(t)$ and (d) $\mathcal{F}_{\text{norm}}(t)$, and (e) time to be in formation.



(a)

(c)

(b)

(d)

(e)

Fig. 14. Effect of $n \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ faulty robots affected by a partial RAB failure type on the performance of a swarm composed of 75 foot-bots: box plots with least square regression fit for (a) $\mathcal{E}(t)$ and (b) $\mathcal{F}_{\text{norm}}(t)$, correlation contour plot for (c) $\mathcal{E}(t)$ and (d) $\mathcal{F}_{\text{norm}}(t)$, (e) and time to be in formation.

However, we found that the swarm normalized weighted-distance distribution quality is almost unaffected by this type of fault, as Fig. 13(b) suggests. This can be largely confirmed in the correlation contour plots illustrated in Fig. 13(c) and (d), respectively, where more increasing high correlated values by high yellow areas can be observed in the results of $\mathcal{E}(t)$ than that observed in the results of $\mathcal{F}_{\text{norm}}(t)$. However, the overall performance of the swarm can be still tolerant to the partial motion failure type, since the maximum $\mathcal{E}(t)$' median value achieved when half of the robots is faulty is nearly 10 cm. Another interesting result when analyzing the time taken for the swarm to be in the formation for this case study is shown in Fig. 13(e). The figure highlights that a slight increase in the TIF is observed with regards to the number of robots affected by a partial motor failure. However, the rate of this increase is almost small.

On the other hand, it seems that the overall performance of the robot swarm is slightly more affected by a partial RAB failure while compared to a partial motor failure. From Fig. 14, it can be clearly seen that the three metrics are affected by the increase of the number of robots being affected by this kind of fault [see the increase of $\mathcal{E}(t)$ and TIF(t) in Fig. 14(a) and (e), respectively, and the slight decrease of $\mathcal{F}_{\text{norm}}(t)$ in Fig. 14(b)]. Better visualization of the correlation between the % of faulty robots is shown in the corresponding correlation contour plots in Fig. 14(c) and (d), respectively. The contour plots make it clear that $\mathcal{E}(t)$ is being more affected by a partial RAB fault type while compared to the effect of such failure mode in the performance of $\mathcal{F}_{\text{norm}}(t)$. However, this fact is still tolerant of the overall performance of the swarm in achieving self-organized cubic-spline-based patterns.

V. Conclusion

The self-organizing robot swarm system into cubic-spline-based patterns can be useful in scenarios, such as smoothly avoiding obstacles during movement or exploring and covering areas with complex curved patterns. Providing effective formation controls for such a challenge was the motivation behind this work. Here, we essentially relied on a set of particular cubic-spline kernel functions (M_4), which is largely applied in the SPHs method for simulating interactive fluid dynamics. This method is specifically used here to interpolate the density of the robots in the swarm as a physical quantity of information moving with the robots. This information should be used then to weigh the distances toward robot neighbors for the purpose to build an AVM among each robot's three available neighbors having the smallest weighted distances. With this technique, we showed that the swarm with different sizes was able to emerge cubic-spline patterns even in the presence of obstacles in the arena. The model also exhibited a good efficiency against noisy sensors, scaled well to different swarm sizes, and can be tolerant to both partial motion failure type and partial RAB fault type. We have also shown that by implicating other kernels of the SPH technique, specifically those based on the Schoenberg *B*-spline functions [11], such as M5 (quartic) and M6 (quantic) kernels can be applied to emerge different *B*-spline formations.

In future work, we aim further to investigate in more detail the effect of the Schoenberg *B*-spline functions on the type of formations that could emerge from the approach both in the presence and absence of obstacles. We also aim to study the use of other kernel functions that differ from the SPH approach (i.e., the Gaussian kernel functions) as well as the possible applications of the approach in real scenarios with real swarm robotics platforms.

References

[1] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, E. Bonabeau, and G. Theraula, *Self-Organization in Biological Systems*, vol. 7. Princeton, NJ, USA: Princeton Univ. Press, 2003.

[2] L. Bai, M. Eyiyurekli, and D. E. Breen, "An emergent system for self-aligning and self-organizing shape primitives," in *Proc. 2nd IEEE Int. Conf. Self-Adapt. Self-Org. Syst.*, Venezia, Italy, 2008, pp. 445–454.

[3] T. Saha and M. Galic, "Self-organization across scales: From molecules to organisms," *Philos. Trans. Roy. Soc. B, Biol. Sci.*, vol. 373, no. 1747, 2018, Art. no. 20170113.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, 2013.

[5] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *Int. J. Comput. Appl.*, vol. 126, no. 2, pp. 31–37, 2015.

[6] S. Hauert, "Swarm engineering across scales: From robots to nanomedicine," in *Proc. Artif. Life Conf.*, 2017, pp. 11–12.

[7] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robot. Auton. Syst.*, vol. 91, pp. 83–100, May 2017.

[8] E. Ferrante, A. E. Turgut, N. Mathews, M. Birattari, and M. Dorigo, "Flocking in stationary and non-stationary environments: A novel communication strategy for heading alignment," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2010, pp. 331–340.

[9] E. Ferrante, W. Sun, A. E. Turgut, M. Dorigo, M. Birattari, and T. Wenseleers, "Self-organized flocking with conflicting goal directions," in *Proc. Eur. Conf. Complex Syst.*, 2013, pp. 607–613.

[10] E. Ferrante, A. E. Turgut, A. Stranieri, C. Pinciroli, M. Birattari, and M. Dorigo, "A self-adaptive communication strategy for flocking in stationary and non-stationary environments," *Nat. Comput.*, vol. 13, no. 2, pp. 225–245, 2014.

[11] D. J. Price, "Smoothed particle hydrodynamics and magnetohydrodynamics," *J. Comput. Phys.*, vol. 231, no. 3, pp. 759–794, 2012.

[12] D. Violeau and B. D. Rogers, "Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future," *J. Hydraul. Res.*, vol. 54, no. 1, pp. 1–26, 2016.

[13] C. Pinciroli *et al.*, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intell.*, vol. 6, no. 4, pp. 271–295, 2012.

[14] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, "A distance weighted-based approach for self-organized aggregation in robot swarms," in *Proc. IEEE 5th Int. Conf. Elect. Eng. Boumerdes (ICEE-B)*, Boumerdes, Algeria, 2017, pp. 1–6.

[15] T. Vicsek and A. Zafeiris, "Collective motion," *Phys. Rep.*, vol. 517, nos. 3–4, pp. 71–140, 2012.

[16] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Techn.*, 1987, pp. 25–34.

[17] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, p. 1226, 1995.

[18] M. Ballerini *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proc. Nat. Acad. Sci.*, vol. 105, no. 4, pp. 1232–1237, 2008.

[19] M. Ballerini *et al.*, "Empirical investigation of starling flocks: A benchmark study in collective animal behaviour," *Anim. Behav.*, vol. 76, no. 1, pp. 201–215, 2008.

[20] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[21] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *Math. Problems Eng.*, vol. 2014, Jun. 2014, Art. no. 205759.

[22] R. Haghighi and C. C. Cheah, "Self-aggregation in multi-agent shape control," in *Proc. IEEE Conf. Robot. Autom. Mechatronics*, Singapore, 2010, pp. 212–217.

[23] S. P. Hou and C. C. Cheah, "Multiplicative potential energy function for swarm control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 4363–4368.

[24] R. Haghighi and C.-C. Cheah, "Multi-group coordination control for robot swarms," *Automatica*, vol. 48, no. 10, pp. 2526–2534, 2012.

[25] H. Zhao, J. Wei, S. Huang, L. Zhou, and Q. Tang, "Regular topology formation based on artificial forces for distributed mobile robotic networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2415–2429, Oct. 2019.

[26] H. Zhao, H. Liu, Y.-W. Leung, and X. Chu, "Self-adaptive collective motion of swarm robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1533–1545, Oct. 2018.

[27] K. D. Listmann, M. V. Masalawala, and J. Adamy, "Consensus for formation control of nonholonomic mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3886–3891.

[28] Z. Sun, Y. Xia, and X. Na, "Consensus-based formation control with dynamic role assignment and obstacle avoidance," *IMA J. Math. Control Inf.*, vol. 34, no. 1, pp. 311–335, 2017.

[29] X. Fu, H. Wang, J. Pan, and X. Gao, "A distributed formation control method of swarm UAVs based on artificial potential field and consensus strategy," in *Proc. IEEE Aust. New Zealand Control Conf. (ANZCC)*, Auckland, New Zealand, 2019, pp. 210–214.

[30] X. Fu, J. Pan, H. Wang, and X. Gao, "A formation maintenance and reconstruction method of UAV swarm based on distributed control with obstacle avoidance," in *Proc. IEEE Aust. New Zealand Control Conf. (ANZCC)*, 2019, pp. 205–209.

[31] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Auton. Robots*, vol. 17, nos. 2–3, pp. 137–162, 2004.

[32] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intell.*, vol. 2, nos. 2–4, pp. 97–120, 2008.

[33] H. Çelikkanat and E. Şahin, "Steering self-organized robot flocks through externally guided individuals," *Neural Comput. Appl.*, vol. 19, no. 6, pp. 849–865, 2010.

[34] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo, "Self-organized flocking with a mobile robot swarm: A novel motion control method," *Adapt. Behav.*, vol. 20, no. 6, pp. 460–477, 2012.

[35] C. Kownacki and D. Ołdziej, "Fixed-wing UAVs flock control through cohesion and repulsion behaviours combined with a leadership," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 1, p. 36, 2016.

[36] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, "Self-organization in aggregating robot swarms: A DW-KNN topological approach," *Biosystems*, vol. 165, pp. 106–121, Mar. 2018.

[37] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annu. Rev. Astron. Astrophys.*, vol. 30, no. 1, pp. 543–574, 1992.

[38] B. Khaldi and F. Cherif, "A virtual viscoelastic based aggregation model for self-organization of swarm robots system," in *Proc. Conf. Towards Auton. Robot. Syst.*, 2016, pp. 202–213.

[39] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, "Improving robots swarm aggregation performance through the minkowski distance function," in *Proc. IEEE 6th Int. Conf. Mechatronics Robot. Eng. (ICMRE)*, Barcelona, Spain, 2020, pp. 87–91.

[40] I. Navarro and F. Matia, "A proposal of a set of metrics for collective movement of robots," in *Proc. Workshop Good Exp. Methodol. Robot.*, 2009, pp. 1–6.

[41] F. Harrou, B. Khaldi, Y. Sun, and F. Cherif, "Monitoring robotic swarm systems under noisy conditions using an effective fault detection strategy," *IEEE Sensors J.*, vol. 19, no. 3, pp. 1141–1152, Feb. 2019.

Belkacem Khaldi received the Dipl.-Ing. degree in informatics from Mohamed Khider Biskra University, Biskra, Algeria, in 2001, the M.Sc. degree in image synthesis and artificial life, and the Ph.D. degree in informatics from Mohamed Khider Biskra University, Biskra, Algeria, in 2012 and 2018, respectively.

Since September 2019, he has been as an Assistant Professor with the École Supérieure en Informatique, 08 Mai 1945, Sidi Bel Abbés, Algeria. He was a Software Developer with Sonatrach Company, Hydra, Algeria, from 2006 to August 2019. He is particularly working on studying a number of problems in swarm robotics, including geometric formation and aggregating patterns. His main research focuses on biological swarms, multiagents, swarm robotics, swarm intelligence, and artificial intelligence. He is also interested in developing machine learning techniques to monitor swarm robotics systems and rediscover governing interaction rules among swarm components.

Fouzi Harrou (Member, IEEE) received the M.Sc. degree in telecommunications and networking from the University of Paris VI, Paris, France, in 2006, and the Ph.D. degree in systems optimization and security from the University of Technology of Troyes (UTT), Troyes, France, in 2010.

He was an Assistant Professor with UTT for one year and the Institute of Automotive and Transport Engineering, Nevers, France, for one year. He was also a Postdoctoral Research Associate with the Systems Modeling and Dependability Laboratory, UTT, for one year. He was a Research Scientist with the Chemical Engineering Department, Texas A&M University at Qatar, Doha, Qatar, for three years. He is actually a Research Scientist with the Division of Computer, Electrical, and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. He has coauthored the book *Statistical Process Monitoring Using Advanced Data-Driven and Deep Learning Approaches: Theory and Practical Applications* (Elsevier, 2020). His current research interests include statistical decision theory and its applications, fault detection and diagnosis, and deep learning.

Foudil Cherif was born in Algeria, in 1962. He received the first degree as an Engineer in computer science from the University of Constantine, Constantine, Algeria, in 1985, the M.Sc. degree in computer science from Bristol University, Bristol, U.K., in 1989, and the Ph.D. degree in computer science from the University of Mohamed Khider, Biskra, Algeria, in 2006.

He is currently a Professor with the Department of Computer Science, University of Mohamed Khider, where he is the Head of team behavioral animation with LESIA Laboratory. His research focuses primary on behavioral animation, crowd simulation, swarm robotics, cryptography, autonomous agents, artificial life, and artificial intelligence.

Ying Sun received the Ph.D. degree in statistics from Texas A&M University at Qatar, Doha, Qatar, in 2011.

She held a two-year Postdoctoral Research position with the Statistical and Applied Mathematical Sciences Institute, Durham, NC, USA, and the University of Chicago, Chicago, IL, USA. She was an Assistant Professor with Ohio State University, Columbus, OH, USA, for a year before joining KAUST, in 2014. At KAUST, she established and leads the Environmental Statistics Research Group, which works on developing statistical models and methods for complex data to address important environmental problems. She has made original contributions to environmental statistics, in particular, in the areas of spatiotemporal statistics, functional data analysis, visualization, computational statistics, with an exceptionally broad array of applications.

Dr. Sun received two prestigious awards, such as the Early Investigator Award in Environmental Statistics presented by the American Statistical Association and the Abdel El-Shaarawi Young Research Award from the International Environmetrics Society.