BioSystems 165 (2018) 106–121

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Self-organization in aggregating robot swarms: A DW-KNN topological approach

Belkacem Khaldi[a], Fouzi Harrou[b,*], Foudil Cherif[a], Ying Sun[b]

[a] LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, R.P. 07000 Biskra, Algeria
[b] King Abdullah University of Science and Technology (KAUST) Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, Thuwal 23955-6900, Saudi Arabia

ARTICLE INFO

Article history:
Received 21 July 2017
Received in revised form 1 December 2017
Accepted 22 January 2018
Available online 2 February 2018

Keywords:
Swarm robotics
Self-organized aggregation
Smoothed Particles Hydrodynamics (SPH)
Virtual viscoelastic model
Distance-Weighted KNN

ABSTRACT

In certain swarm applications, where the inter-agent distance is not the only factor in the collective behaviours of the swarm, additional properties such as density could have a crucial effect. In this paper, we propose applying a Distance-Weighted K-Nearest Neighbouring (DW-KNN) topology to the behaviour of robot swarms performing self-organized aggregation, in combination with a virtual physics approach to keep the robots together. A distance-weighted function based on a Smoothed Particle Hydrodynamic (SPH) interpolation approach, which is used to evaluate the robot density in the swarm, is applied as the key factor for identifying the K-nearest neighbours taken into account when aggregating the robots. The intra virtual physical connectivity among these neighbours is achieved using a virtual viscoelastic-based proximity model. With the ARGoS based-simulator, we model and evaluate the proposed approach, showing various self-organized aggregations performed by a swarm of *N* foot-bot robots. Also, we compared the aggregation quality of DW-KNN aggregation approach to that of the conventional KNN approach and found better performance.

1. Introduction

Aggregation is a fundamental behaviour that is observed in many biological organisms, such as social insects and group-living animals (Camazine et al., 2002). It is an important requirement for animal societies to accomplish complex swarming behaviours collectively. This can be achieved using external gradients, such as humidity for woodlice (Broly et al., 2012, 2013) or temperature for honeybees (Kernbach et al., 2009); this kind of aggregation is mostly known as cue-based aggregation (Arvin et al., 2014b). It can be also achieved in self-organized manner, where the aggregation process is enabled without any external cues (Camazine et al., 2002). Fascinating examples of such behaviour can be observed in bird flocks, fish schools, and mammal herds (Hemelrijk and Hildenbrandt, 2012).

On the basis of these biological studies, aggregation is also a matter of interest in various swarm robotics studies. It is considered as a fundamental task that allows robot swarms to perform complex tasks, such as collective movement, self-assembly, and pattern formation, or to exchange information.

This study takes inspiration from the topological distance approach revealed in studies of birds flocking and fish schooling. Ballerini et al. (2008a,b) showed empirically that the unpredictable, amazingly complex patterns formed by birds emerge from a topological distance approach rather than a metric distance approach, meaning that the birds only interact with their nearest six or seven neighbours rather than all of the neighbours in their field of vision.

Computer simulations predict that a significantly higher cohesion of the aggregation is achieved using a topological interaction rather than the standard metric one. Similarly, empirical results show that fish and elephants, for example, interact with only three or four neighbours (Niizato et al., 2014).

In swarm robotics, few models have applied such topological approaches to the study of collective behaviours. Some researchers proposed selecting strategies before interacting with nearby robots so that only a fixed number of neighbours are used (Lee and Chong, 2008; Ercan et al., 2010). For example, Lee and Chong (2008) proposed a bio-inspired flocking control based on selecting two neighbors for team maintenance and local interactions. Ercan et al. (2010) introduced a regular tetrahedron formation strategy for selecting the three neighbours that forms the best tetrahedron to ensure formation. In our previous work (Khaldi and Cherif, 2016a), we studied self-organized patterns that emerge from an approach based on the intra-virtual physical connectivity among

[*] Corresponding author.
E-mail address: fouzi.harrou@kaust.edu.sa (F. Harrou).

https://doi.org/10.1016/j.biosystems.2018.01.005
0303-2647/© 2018 Elsevier B.V. All rights reserved.

the K-Nearest Neighbours (KNN). However, most of the topologies used in these studies rely on simple distances constraints. Relying on such constraints is not always the perfect solution in many real-world scenarios, such as when the network nodes hold supplementary properties that differentiate them from others. In such cases, the interactions between the nodes are best described by weighted connections instead of simple ones.

In this paper, a new topological distance approach is proposed to study self-organization in aggregating robot swarms. We use the density of robots in the swarm as an additional property and define a new neighbouring relationship based on the Distance-Weighted KNN (DW-KNN) (Section 4.2.2). In certain applications, such as driving a large number of robots from one area to another, the density of the robots could play the primary role while the inter-robot distance would play only a secondary role as a metric useful for proximity control or collision avoidance. With the DW-KNN topology, virtual physical connections are dynamically created and destroyed among the weighted KNN rather than the unweighted KNN. To achieve this topology, we introduce a Smoothed Particle Hydrodynamic (SPH) density-based approach (Section 4.2.1) to evaluate the density of the robots in the swarm and to weight the distances among neighbours. SPH is a mesh-free Lagrangian method that is widely used for many diverse applications, including astrophysics, geophysics, engineering, and the film and computer games industries. The method is based on an interpolation technique that approximates physical quantities that are moving with particles (Violeau and Rogers, 2016); here we approximate the density as a physical quantity moving with the robots. As in Khaldi and Cherif (2016a,b), we use a virtual viscoelastic-based proximity model (Section 4.1) as the intra-virtual physics connectivity among the neighbouring robots; the virtual viscoelastic connections are Voigt links, and each link can be represented by a purely elastic spring and a purely viscous damper connected in parallel. We also incorporate a repulsive primitive to avoid obstacles and collisions with the robots that are not virtually linked together (Section 4.4). The overall control model (Section 5) with a swarm of foot-bot robots is designed and evaluated using the ARGoS simulator (Pinciroli et al., 2012). The results obtained in (Section 6) by simulating up to 150 foot-bot robots in both absence and presence of obstacles show the variation that emerged in the self-organized aggregation from the execution of the overall control model. The performance of the model is studied using three different metrics (Section 7.1): the distance-weighted distribution quality (F_W), the aggregation quality (F_{ag}), and the Averaged Mean Distance Error (*AMDE*) metric. With these metrics, we analyze the aggregation performance in normal circumstances (Section 7.2), and we later assess how the aggregation performance is affected when the readings of the robots' range and bearing sensors are corrupted by noise (Section 7.3). Furthermore, in Section 8, we provide comparisons of the proposed approach with the conventional KNN approach and showed that we achieve better results.

2. Related works

Aggregation is a current focus of swarm robotics, and several studies on both cue-based aggregation and self-organized aggregation have been completed in the last two decades.

2.1. Works on cue-based aggregation

In one study of cue-based aggregation by Kube and Zhang (1993), a light source was used to aggregate a robot system around an object and to transport the object in a collaborative manner to another goal; by following simple behaviours (e.g., to find light and follow light), the aggregation task was completed with no explicit communication mechanisms. (Holland and Melhuish, 1997) proposed a method based on an infra-red (IR) cue to regulate the size of an aggregate created by a robotics system that successfully allowed each robot to approximate the aggregate size and decide to join or leave the aggregate accordingly. Mermoud et al. (2010) used a cue-based aggregation approach to address the problem of collective decision-making in swarm robotics system. In that study, the robots applied a probabilistic aggregation mechanism that first allowed them to aggregate in a good or bad location; then, based on the status of the location, the robots made a collective decision whether to keep or destroy the aggregate in that location.

Recently, one of the most successful cue-based aggregation models took inspiration from the collective behaviour of honeybees, which prefer gathering where the temperature is 36°C. The BEECLUST model proposed by Kernbach et al. (2009) was the first algorithm that mimicked this behaviour; a gradual light source was used to generate clustering behavior in a swarm robotics system. It was proven to act robustly in many researches (Bodi et al., 2012, 2015; Arvin et al., 2014a, 2016). Further, different variations of the model have been suggested to increase the performance of the aggregation process. For instance, Arvin et al. (2011) proposed a new aggregation algorithm in which a dynamic velocity and a comparative waiting time were introduced to the original BEECLUST model, which contributed to a significant improvement in the aggregation time. Furthermore, a comparison between the original BEECLUST algorithm and two modified versions – called the vector averaging algorithm and the naive algorithm – showed that both the vector averaging and naïve algorithms outperformed the original BEECLUST model, and revealed that noise has less impact in the vector averaging method than the naïve one (Arvin et al., 2014a).

Later, those authors introduced a fuzzy-based aggregation approach to enhance the performance of BEECLUST in both computer-based simulations and real robot swarms (Arvin et al., 2014b). Wahby et al. (2016), proposed another adaptive variant of BEECLUST, where the original algorithm was extended to adapt automatically to any light conditions. Recently, Bodi et al. (2012) proposed a model called ODOCLUST, which incorporated odometry as an additional capability to BEECLUST. The ODOCLUST variant achieved a fast and accurate aggregation without requiring high-fidelity odometry.

2.2. Self-organized aggregation works

In this paper, we are specifically interested in studying self-organized aggregation, in which robot swarms achieve aggregation using simple local interaction rules among individuals. Different approaches have been proposed to address the problem of self-organized aggregation in swarm robotics. Garnier et al. (2008) adopted a probabilistic approach, inspired by the cockroach model of Jeanson et al. (2005), to achieve aggregation using a swarm of 20 physical Alice robots in homogeneous environments. A similar work by Correll and Martinoli (2011) showed that when using probabilistic aggregation rules, a minimum combination of communication range and locomotion speed was needed to achieve a single aggregate cluster. Soysal and Sahin (2005) suggested a probabilistic aggregation method in which a state-finite machine was used to combine a set of simple behaviours that included avoiding an obstacle, approaching, repelling, and waiting.

In other studies, self-organized aggregation was studied using deterministic approaches. In these methods, the robots build a connected visibility graph, and ensure that the graph is permanently maintained. Ando et al. (1999) used an algorithm in this sense to study aggregation in a group of mobile robots with a limited sensing range. Later, the algorithm was generalized by Ando et al. (1999) to achieve an aggregation in arbitrarily high dimensions. The forma-

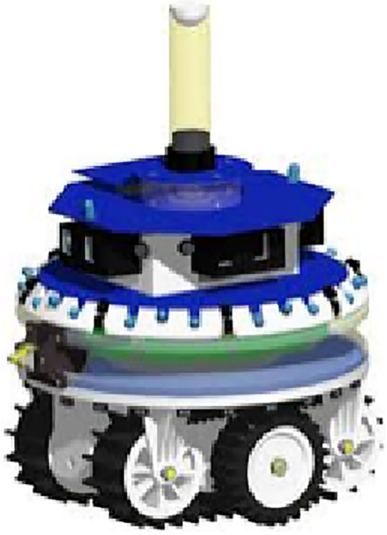


Fig. 1. The foot-bot robot, CAD model (Pini et al., 2009).

tion of the graph in these algorithms was based on the assumption that the robots were able to measure both the range and the angle of their neighbours. However, Gordon et al. (2004) were able to achieve such an aggregation using only the angle measurement of the robot's neighbours. The aggregation performance of this last algorithm was later improved, by introducing an additional, crude range-sensing capability for differentiating whether neighbouring robots were near or far in Gordon et al. (2008). In another study, De Gennaro and Jadbabaie (2006) used the Laplacian matrix to allow each robot to build its own proximity graph. The Laplacian matrix is an algebraic representation of a graph, through which many useful graph properties can be found (Merris, 1994). The related control was fully decentralized, and simulated results demonstrated that the model was effective and even increased the connectivity of the entire swarm.

In some works, self-organized aggregation models have been approached using artificial evolution techniques. For instance, aggregation with simple robots, called s-bots, was studied by Trianni et al. (2003). In this study, general solutions to the aggregation problem were produced using an evolutionary robotics mechanism. The method was able to produce clustering behaviours with both static and dynamic behavioural strategies. With that model, Dorigo et al. (2004) revealed that effective evolved controllers could be achieved for both aggregating and coordinated motion behaviours in a swarm of s-bots. In a similar setup, Soysal et al. (2007) investigated the effects of a number of parameters, such as the robots' number, the size of arena, and the run time. In another study, Gauci et al. (2014a) proposed two algorithms – a reactive controller with no memory and a recurrent controller with memory – to study aggregation in a swarm of e-puck robots. The algorithms were based on a classical evolutionary programming technique, and used a simple binary sensor with a sensing range that proved sufficient to achieve an error-free aggregation. Results from both the simulation and experiments showed that aggregation toward one cluster was successfully achieved. However, a sufficiently long range in the binary sensor was needed to achieve an accurate aggregation.

Other aggregation studies approached a vision based on virtual physics methods. Gasparri et al. (2012a) adopted an attractive/repulsive virtual force model to study aggregation in a swarm of multi-robot systems based on local interaction. This model was later extended to cope with actuator saturation by Gasparri et al. (2012b) and to integrate obstacle avoidance by Leccese et al. (2013). In most of the works cited above, the inter-robot distance is the only factor taken into account when interconnecting the robots. However, in certain applications where supplementary factors such as the density of the robots could emerge as important, a robot's density plays a more crucial role than the inter-robot distance. In this context, the related methodologies mainly use the SPH model to simulate fluid-like behaviour in a robot swarm. This technique has successfully been used to demonstrate how multi-robotic agents, modelled as a stream of incompressible or compressible fluid are capable of being applied to tasks such as group motion and shape control, group segregation (Zhao et al., 2011), pattern generation (Pimenta et al., 2008, 2013), and unknown area coverage.

Rather than using the SPH model as a virtual fluid force law to design the proximity control of a robot swarm, here we use the SPH model to evaluate the density of the robots in the swarm, then we define a DW-KNN neighbouring topology to aggregate the robots based on this density. Using the DW-KNN-based neighbouring topology, self-organized aggregations were achieved by taking into account two key factors, the inter-robot distance and the robot' density.

3. Method

3.1. Problem formulation

We consider an area surrounded by four walls, containing a swarm of N $(R_1, \ldots, R_N)$ differential two-wheeled mobile robots that are initially distributed in random positions and heading arbitrary directions. We assume that, within a specific range D_r, each robot R_i is sufficiently equipped with sensors to measure the range d_{ij} and the bearing θ_{ij} of its jth neighbour ($j \in \mathcal{N}_i(t)$, where $\mathcal{N}_i(t)$ is the set of neighbours of the robot R_i at a time t). In addition, we consider that the robot can communicate with its neighbours within the same range.

The objective is to study the self-organized aggregations that emerge through a topological neighbourhood approach without using any cues.

3.2. Simulation platform

The simulations in this study were performed on the ARGoS platform (Pinciroli et al., 2012). ARGoS is an open source multi-robot simulator that was first developed within the EU-funded Swarmanoid project[1] to study tools and control strategies for heterogeneous swarms of robots. Now, it is one of the most-used simulation platforms in many researches and projects that are dedicated to synthesis swarming behaviour controls. ARGoS can simulate large-scale, heterogenous swarms of robots in real time. Moreover, it comes with all the tools needed for the development cycle of robot control code, from design to validation on real robots; therefore, there is no difference between coding for simulation or reality (Pinciroli et al., 2012). ARGoS has a built-in models for several well-known robot such as the foot-bot, e-puck, kilobot, and fly-bots.

The robots used in our experimental simulations are foot-bots (see Fig. 1), two-wheeled differential mobile robots developed by Bonani et al. (2010) that come with the actuators and sensors required by our control model. In our experiments, we use the following.

- A range and bearing perception and communication device (called RAB), that allows the robot communicating with its neigh-

[1] http://www.swarmanoid.org.

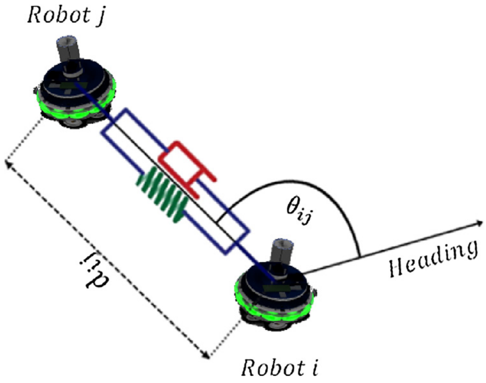


Fig. 2. A virtual viscoelastic interaction model between two robots modeled as a *Voigt* connection.

bours, as well as identifying their relative range and bearing measures.

- Proximity sensors that allow detecting objects around the robot. The foot-bot robot is equipped by 24 sensors, which are uniformly distributed in a ring around the robot body.
- Two wheels actuators to independently control the forward speed of the left and the right wheels of the robot.

The body of the foot-bot is about 8.5 cm wide and 14.7 cm tall. The inter-wheel distance is 0.14 cm. The velocities of the left and right wheels along the ground can be set independently to a maximum speed of [13, 13] cm/s.

4. Robot model

In this study, the robot model was built following an artificial physics design approach (Spears et al., 2004). In this method, the robots sense and respond to virtual forces that are driven by regular physics laws. Here, the robot was based on our previously proposed model (Khaldi et al., 2017; Khaldi and Cherif, 2016a,b), where the robot is subjected to the virtual force vector

$$\hat{a}_i = \hat{p}_i + \hat{r}_i. \tag{1}$$

The proximity control vector, $\hat{p}_i$, is used to encode the attracting rules. The repulsive control vector, $\hat{r}_i$, is used to encode the repulsing rules.

4.1. Proximity model

The proximity control is the crucial model in our study; it was used to aggregate the robots together while maintaining a certain distance between them. To achieve this model, a topological neighbourhood strategy is applied first by each robot R_i to decide which neighbours among those available are taken into account when arranging the robots. Then, a mesh of virtual viscoelastic links is built between the robots of the topological neighbourhood by modelling the sensing capabilities using *Voigt* interactions (see Fig. 2) (Khaldi and Cherif, 2016a,b). The proximity model of each robot R_i is computed as

$$\begin{aligned}\hat{p}_i = &\sum_{j\in\ \mathcal{T}_i(t)} (k^s_{ij}(d_{ij}-d_0)\hat{d}_{ij} + k^d_{ij}(v_i - v_j)),\\ \text{with}\ \ k^s_{ij} = &\frac{k_s}{\sqrt{d_{ij}}},\quad k^d_{ij} = k_d\sqrt{k^s_{ij}},\quad \text{and}\quad i \neq j,\end{aligned} \tag{2}$$

where $\mathcal{T}_i(t) \in \mathcal{N}_i(t)$ is the set of topological neighbours at time t (more details about how the robot identifies this set is discussed in the next subsection), and k^s_{ij} and k^d_{ij} are the spring and the damping coefficients, respectively. The values of these coefficients are dynamically changing with regards to the actual distance d_{ij} between a robot R_i and a robot R_j, and the two gain constants, k_s and k_d. The equilibrium length of the spring is d_0, and $\hat{d}_{ij}$ is the unit vector indicating the direction of the force of the virtual spring. Finally, v_i and v_j are the forward velocities of the robots R_i and R_j, respectively.

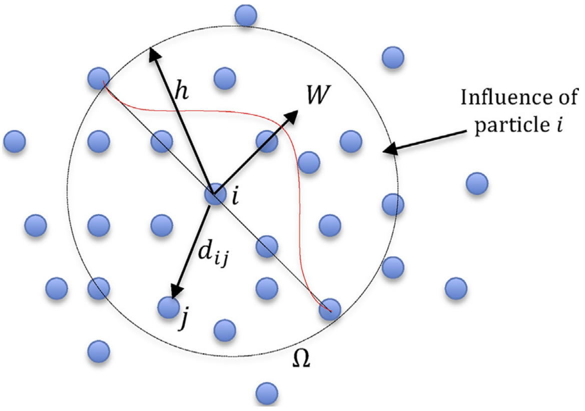


Fig. 3. A schematic representation of SPH particle interactions within the influence domain governed by the kernel function.

4.2. Topological neighborhood approach

We propose the DW-KNN method as a topological approach to identify $\mathcal{T}_i$. The weighted distances are computed as follows:

- First, a robot R_i computes its density ρ_i based on an SPH density estimation technique, which should be immediately communicated to its neighbours.
- Second, upon receiving the densities (ρ_j) from neighbours, a weighted-distance function w_{ij} is applied to weight the distances to the neighbours.
- Finally, based on w_{ij}, the set $\mathcal{T}_i$ is identified by sorting the neighbours in order of the nearest K-weighted distances, where K refers to how many neighbours are taken into account.

The SPH density estimation technique and the weighted-distance-based function we use in our study are discussed in the following subsections.

4.2.1. SPH density estimation technique

SPH is a mesh-free Lagrangian method, in which the state of the simulated system is represented by employing a finite set of disorder discrete particles in a way that makes both a fixed order to organize the particles and a generated mesh to represent the connectivity of the particles unnecessary (Violeau and Rogers, 2016). SPH has been firstly introduced in computational astrophysics studies and is applied in simulating compressible flow problems (Price, 2012). One of the main features of the SPH technique is that at any given point in the simulation domain Ω, a property of a particle i can be approximated relying on a summation of an interpolation kernel function W with h as the smoothing length (Pimenta et al., 2013). A schematic representation of this system is presented below in Fig. 3

In the SPH approach, interpolation is used to approximate physical quantities that are moving with particles; here we approximate the density as a physical quantity moving with the robots. The robot density ρ_i is evaluated as the weighted sum of distances over its proximate robots within a particular range $D_r = 2h$ (Zhao et al., 2011):

$$\rho_i = \sum_{j \in \mathcal{N}_i} W(d_{ij}, h), \tag{3}$$

The weight functions used in this work are the $M4$ cubic spline functions truncated at $2h$ (Price, 2012):

$$W(d_{ij}, h) = \sigma \begin{cases} \frac{1}{4}(2-q)^3 - (1-q)^3, & 0 \le q < 1 \\ \frac{1}{4}(2-q)^3, & 1 \le q < 2 \\ 0, & q \ge 2 \end{cases} \tag{4}$$

where $q = d_{ij}/h$ and $\sigma = 10/(7\pi h^2)$ is a normalized constant. The computed density is communicated to the neighbours of the robots.

4.2.2. Distance-Weighted K-Nearest Neighbours

DW-KNN is a very popular acronym in machine learning. It is a classification method that is specifically used to assign a label to a new query based on weighting closer neighbours more heavily, according to their distances from the query (Gou et al., 2012). Taking inspiration from this technique, and upon receiving the SPH densities from the neighbours, a robot R_i weights the distances to its neighbours as

$$w_{ij} = \rho_j d_{ij}, \tag{5}$$

where ρ_j denotes the density received from the jth neighbour. Further, the robot builds a DW-KNN connectivity, $\mathcal{T}_i(t)$, by sorting the neighbours in order of the nearest K-weighted distances, where $K \in \{1, 2, \ldots, N-1\}$ refers to how many neighbours are taken into account. The cardinality of the neighbourhood, $|\mathcal{T}_i(t)|$ is K, and $j \in \mathcal{T}_i(t) : w_{ij} \le w_{im}, \forall m \in \mathcal{T}_i(t)$. In the case where $K = N-1$, the mesh is the all-to-all connected network.

By using the function in Eq. (5) where both the distance and the density are applied as equal key factors, a neighbour robot R_j located far away from the robot R_i with a heavy density could have a greater impact than one located near R_i but with a weak density.

4.3. Obstacle/collision avoidance model

This model was used specifically to avoid obstacles encountered in the arena. Since the robots were placed in an inbound environment surrounded by four walls, we denote the set of the four walls and obstacles present in the arena by $\mathcal{O}_i$. In addition, since the robots only interacted with their DW-KNN robots ($\mathcal{T}_i$), the model was also activated to avoid collisions with the remaining neighbouring robots ($\mathcal{M}_i = \mathcal{N}_i \backslash \mathcal{T}_i$). To achieve this model, a repulsive potential field was generated around each robot. The field had a strong influence when a robot was close to the potential field and a decreasing effect as the robot moved further away. We compute the repulsive control vector as follows (Khaldi and Cherif, 2016a,b):

$$\hat{r}_i = \begin{cases} k_r(\frac{1}{L_j} - \frac{1}{L_0})(\frac{1}{L_j^2}), & L_j \le L_0 \\ 0, & \text{elsewhere} \end{cases} \tag{6}$$

where k_r is a scaling constant, L_j ($j \in \mathcal{A}_i = \mathcal{O}_i \cup \mathcal{M}_i$) is the distance to the nearest obstacle or the nearest robot, and L_0 is the obstacle influence threshold.

4.4. Motion Control

At each control step, each robot R_i transforms the total vector, $\hat{a}_i$, into signals that update the forward speeds (v_{l_i} and v_{r_i}) of its left and right wheels, respectively, as follows (Khaldi and Cherif, 2016a,b):

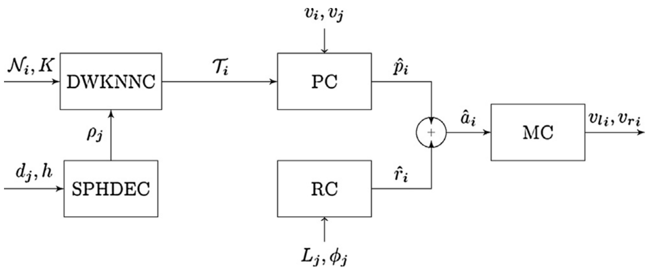


Fig. 4. The overall self-organized aggregation control model.

$$\begin{bmatrix} v_{l_i} \\ v_{r_i} \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & \frac{-b}{2} \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \tag{7}$$

$$\omega_i = k_\omega \left(\frac{180\angle\hat{a}_i}{\pi} \right), \quad v_i = \frac{v_{\max}}{\sqrt{|\omega_i| + 1}}, \tag{8}$$

where v_i and ω_i, respectively, are the forward and the angular speeds of the robot, b is the robot's inter-wheel distance, $\angle\hat{a}_i$ is the angle formed by the x and y components of the vector $\hat{a}_i$, k_ω is a gain constant, and $v_{\max}$ is the maximum forward speed that the robot can reach. The angular speed should be also within the range $[-\omega_{max}, \omega_{max}]$.

5. Implementation of the overall self-organized aggregation control model

A scheme of the overall control model implemented in the foot-bot robot is shown in Fig. 4. The details of the entire algorithm are illustrated in A.

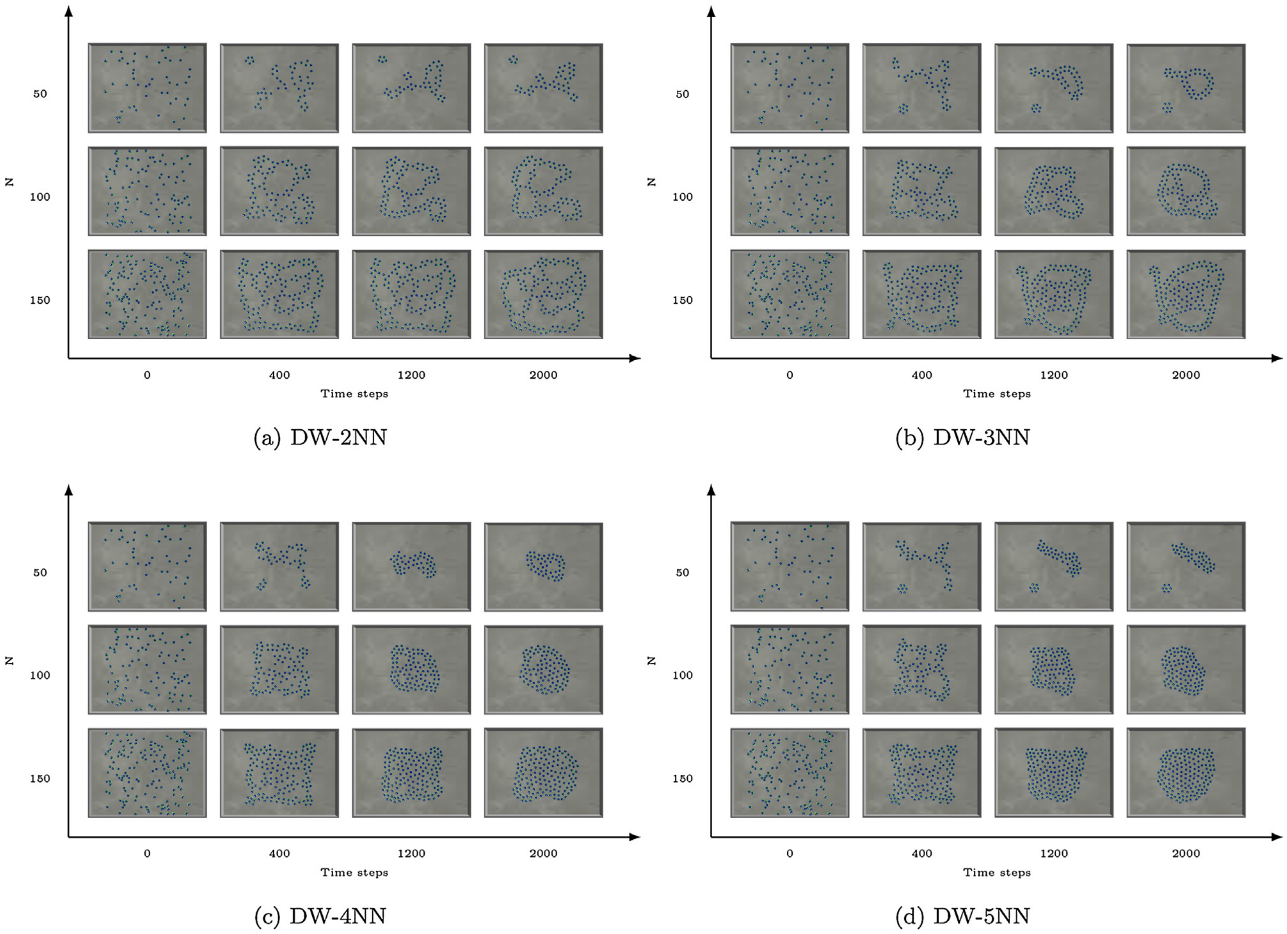
To achieve the SPH Density Estimator Control (SPHDEC), the robot uses the RAB device to measure the distances, d_j, to its neighbours within the perception range, D_r. Using the equations from the SPHDEC model, the foot-bot is able to compute its density, which is immediately sent to its neighbours via the RAB. Since the foot-bot can exchange only 10 bytes of data, the density is scaled down by 10^3 before being sent to the neighbours.

To achieve the DW-KNN Control (DWKNNC), the robot receives the densities of the neighbours and computes its corresponding weighted distances; then, based on the value of K, the robot can determine the DW-KNN.

To recognize the proximity control (PC), the robot uses once more the RAB device to communicate its forward speed to its neighbouring robots, as well as to measure the range and bearing (d_j and θ_j) towards these neighbouring robots. For accomplishing the Repulsive Control (RC), we use proximity sensors to measure the distance (L_j) and the angle (φ_j) of the closet detected obstacle. Finally, based on the resulting vector a_i, the Motion Control (MC) actuates the linear speed of the robot wheels.

6. Simulation results

As mentioned above, all of the simulations in this study were performed using ARGoS platform (Pinciroli et al., 2012). In the following subsections, we illustrate the configuration setup used in the simulation and some of the simulation results, explain the metrics used to evaluate the proposed method, and present some of the results in the context of the metrics.



(a) DW-2NN

(b) DW-3NN

(c) DW-4NN

(d) DW-5NN

Fig. 5. Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology.

Table 1
Parameters and constants used in the simulation.

Parameter	Description	Value
b	Inter-wheels distance	0.14 m
$v_{\max}$	Maximum forward speed	0.10 m/s
ω_{max}	Maximum angular speed	180° s^{-1}
d_0	Equilibrium length of the spring	0.3 m
L_0	Obstacle influence threshold	0.1 m
k_r	Obstacle scaling constant	1.75 force unit
k_d	Damper gain constant	1.25 force unit
k_ω	Angular speed gain	1.5° s^{-1}
k_s	Spring gain constant	1.9 force unit
D_r	Maximum perception range	1.5 m
h	The smoothing length	0.5 m

6.1. Experimental setup

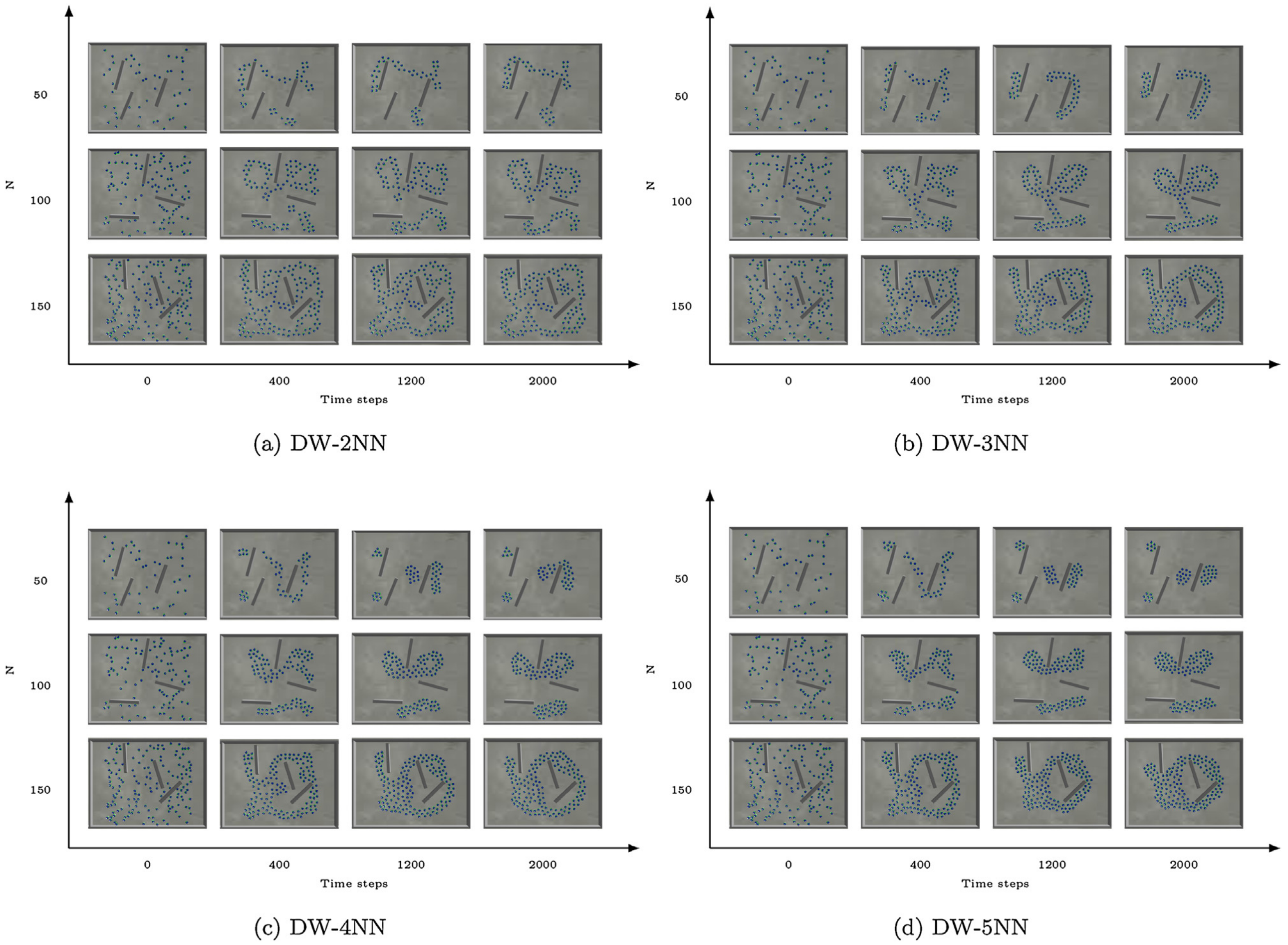
We use a swarm of N foot-bots headed in arbitrary directions that were arbitrarily placed in a $(10 \times 6)\,\mathrm{m}^2$ arena. Noise was added to the range and bearing measurements of the RAB device. Noise was modelled as a Gaussian distribution with zero mean and 0.01 deviation. The loss of packets during communication was also taken into consideration by setting the loss probability to 3%. We conducted three separate experiments with different numbers of foot-bot robots ($N = \{50, 100, 150\}$) for a duration of 2000 time steps (ts) each (1 ts = 0.1 s). Starting from the initial distribution of robots, a set of constants and parameters (see Table 1) were initiated in each experiment.

We demonstrate in Fig. 5 the self-organized aggregations that developed from the execution of the overall control model by a swarm of $N = \{50,\ 100,\ 150\}$ foot-bots robot at diverse time steps ($t = 400$, $t = 1200$, and $t = 2000$) when starting from the initial positions ($t = 0$). Starting from the same initial position, the robot swarm achieved different self-organized aggregations as only the neighbourhood topology was varied. Fig. 5a highlights the results obtained from a DW-2NN topology, Fig. 5b presents the results achieved from a DW-3NN topology, whereas results with DW-4NN and DW-5NN topologies are mentioned in Fig. 5c and Fig. 5d, respectively.

Also, we investigated the DW-KNN approach in presence of obstacles while keeping the same parameters setup of the previous scenario. To do so, three obstacles are randomly placed in the arena. Fig. 6 shows the evolution of the swarm to achieve accurate aggregations while smoothly avoiding obstacles.

It can be seen that in both absence and presence of obstacles, the robots swarm is able to achieve self-organized aggregations via the proposed approach. Moreover, the approach can be useful in scenarios such as driving a large scale of robots from one area to another, while maintaining a connectedness between the robots and avoiding collisions. Here, the connectivity between the neighbours are modelled using virtual viscoelastic links between the DW-KNN, and the distances toward those neighbours are weighted using an SPH density estimation technic where M4 cubic spline functions are applied. This fact could smoothly drive the swarm to emerge cubic based self-organized aggregations as illustrated in snapshots of Figs. 5 and 6.

It can also be noticed that in presence of obstacles, more clusters could emerge compared to a situation without obstacles in the arena. See, for example, snapshots in Fig. 6a, c, and d for the



(a) DW-2NN

(b) DW-3NN

(c) DW-4NN

(d) DW-5NN

Fig. 6. Self-organized aggregation of a swarm of N foot-bots running the DW-KNN topology with existing obstacles.

case when $N = \{50, 100\}$, and snapshots in Fig. 6b for the case of $N = 50$. This is mainly due to the fact that the robot' field vision will be much influenced by the existence of more obstacles, and therefore its range and bearing sensing capabilities can be significantly affected. This means that an obstacle located in a robot' RAB range will blind the vision of a robot to sense and communicate with neighbours. As consequence of an increase of both the size of robots in the swarm and the value of K in the DW-KNN approach results in a decrease of the total number of the clusters that could emerge.

7. Performance analysis

In this section we are interested in studying the performance of our proposed model. Specifically, we evaluate how the robots, relying only on the DW-KNN topology, achieved a self-organized aggregation while maintaining a certain distance between each robot. We also analyze the effect of noise on the model. For that, we use the following metrics to ascertain the attainment of this objective.

7.1. Performance metrics

7.1.1. Distance-weighted distribution quality

We define a new metric to measure the quality of the evolution of the overall weighted distances of the entire swarm. First, the weighted distances averaged over the different robots and neighbours $AWD(t)$ are calculated as follows:

$$AWD(t) = \frac{1}{N.K}\sum_{i=1}^{N}\left(\sum_{j=1}^{K} w_{ij}(t)\right). \tag{9}$$

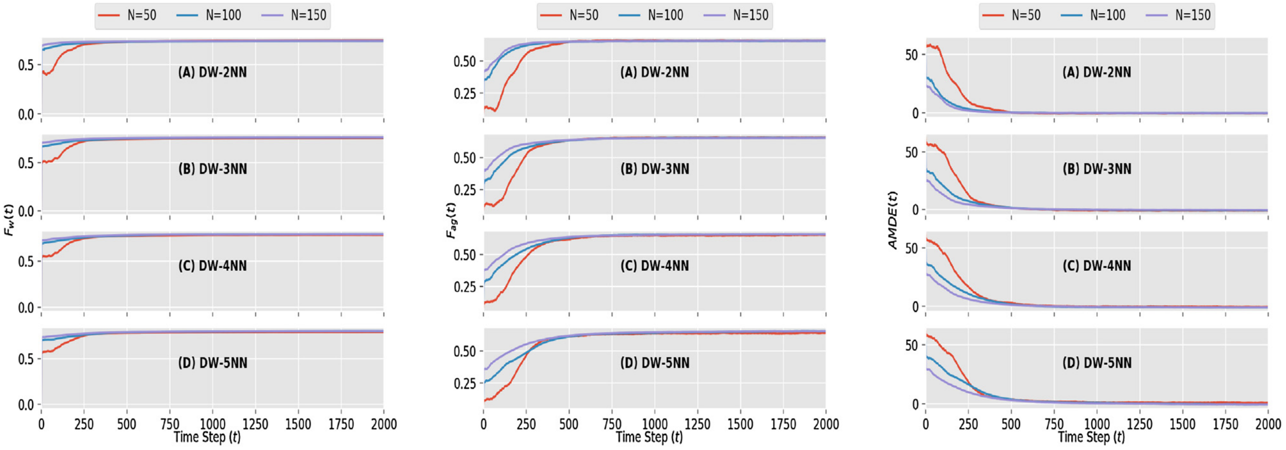
Then the distance-weighted quality metric, $F_w(t)$, is gotten by the following equation:

$$F_w(t) = 1 - \frac{1}{\sqrt{AWD(t) + 1}}. \tag{10}$$

7.1.2. Aggregation quality

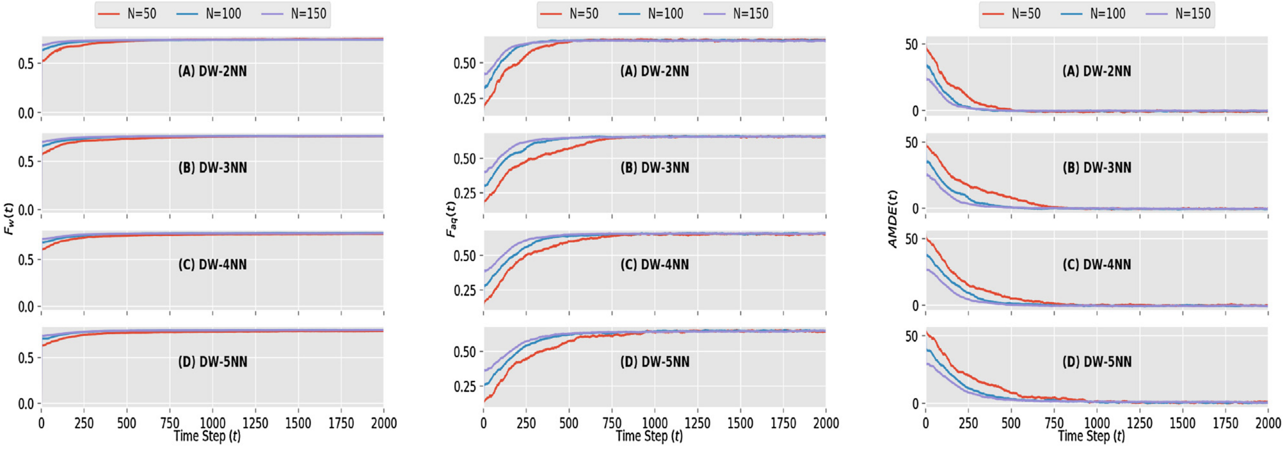
The aggregation quality (Dorigo et al., 2004), $F_{ag}(t)$, is related to the average distance of the robots from their center of mass. To measure this metric, first the distance $c_i(t)$ of each robot R_i from the center of mass of the group at simulation cycle t is computed as

$$c_i(t) = \left\| X_i(t) - \frac{1}{N}\sum_{j=1}^{N} X_j(t) \right\|, \tag{11}$$



(a) (b) (c)

Fig. 7. Performance metrics results in absence of obstacles: (a) $F_w(t)$, (b) $F_{ag}(t)$ and (c) *AMDE*(t) obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 25 runs with different initial configurations of robots.



(a) (b) (c)

Fig. 8. Performance metrics results in presence of obstacles: (a) $F_w(t)$, (b) $F_{ag}(t)$ and (c) *AMDE*(t) obtained from the overall controller implemented on $N = \{50, 100, 150\}$ robots when taking into consideration different DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 25 runs with different initial configurations of robots.

where $X_i(t)$ is the position vector of the ith robot at time step t. This value is used to compute the aggregation quality $Ag_i(t)$ of the ith robot as follows:

$$Ag_i(t) = \begin{cases} 0, & c_i(t) < \tilde{r}(n) \\ \dfrac{R(n) - c_i(t)}{R(n) - \tilde{r}(n)}, & \tilde{r}(n) \leq c_i(t) \leq R(n) \\ 1, & c_i(t) > R(n) \end{cases} \tag{12}$$

where $R(n) = \tilde{r}(n) + 100$, $\tilde{r}(n) = r_s.(\sqrt{n} - 1)$ is defined as an approximation of the radius of the smallest circle that has n robots positioned on its perimeter, and r_s is the radius of a robot (Dorigo et al., 2004).

Then the average aggregation quality, $F_{ag}(t)$, is defined as follows:

$$F_{ag}(t) = \frac{1}{N.K} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} Ag_i(t) \right). \tag{13}$$

7.1.3. Averaged Mean Distance Error

The Averaged Mean Distance Error metric, *AMDE*(t), is defined as the inter-robots distance error averaged over the different robots and neighbours. It is used mainly to measure how well the swarm maintains a certain desired distance between the individual robots as they move together. The metric is calculated as follows:

$$AMDE(t) = \frac{1}{N.K} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} (d_{ij}(t) - d_0) \right). \tag{14}$$

7.2. Analysis in normal circumstances

To evaluate our proposed approach more accurately, we performed 25 runs of each experiment and the experiment duration of each run was set again to 2000 time steps. We plotted the results obtained from the simulation of $N = \{50, 100, 150\}$ foot-bot robots within the DW-2NN, DW-3NN, DW-4NN and the DW-5NN topologies in absence and presence of obstacles. Figs. 7 and 8 show respectively the time evolution of $F_w(t)$, $F_{ag}(t)$, and *AMDE*(t) for

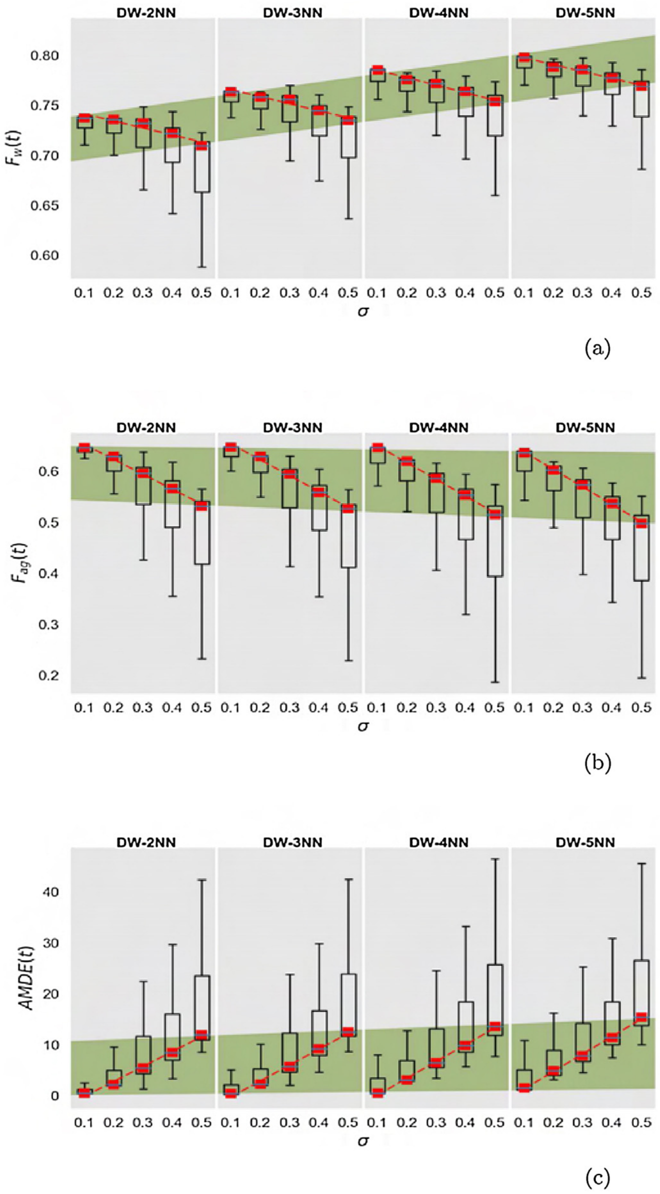


Table 9.a: Coefficients of $f(x)$ for $F_w(t)$

	c_1	c_0
DW-2NN	0.748	-0.007
DW-3NN	0.773	-0.007
DW-4NN	0.793	-0.007
DW-5NN	0.804	-0.007
Olive band	0.004	0.700

Table9 .b: Coefficients of $f(x)$ for $F_{ag}(t)$

	c_1	c_0
DW-2NN	0.682	-0.029
DW-3NN	0.685	-0.031
DW-4NN	0.683	-0.032
DW-5NN	0.672	-0.034
Olive band	-0.002	0.550

Table9 .c: Coefficients of $f(x)$ for $AMDE(t)$

	c_1	c_0
DW-2NN	-2.972	2.922
DW-3NN	-3.272	3.114
DW-4NN	-3.077	3.283
DW-5NN	-2.034	3.430
Olive band	0.062	0.268

Fig. 9. Performance metrics results: (a) $F_w(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on 100 robots when taking into consideration different DW-KNN topologies with different σ. In all plots, the x-axis represents the standard deviation of noise σ. Each box represents values obtained from 25 runs with different initial configurations of robots. The red squares show the median values, and the dashed red lines shows a linear least squares regression fit to these squares; the olive band shows a linear least squares regression fit to the mean of the 5 red squares for each topology. The linear least squares regression functions generated to fit all of these points are of the form $f(x) = c1x + c0$; the corresponding coefficients can be found in the tables to the right of each figure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the four DW-KNN case studies in both absence and presence of obstacles. In all of the plots, the curves show the median values of the 25 runs. Results show that the swarm robot system successfully converged into a stable value of $F_w(t)$, $F_{ag}(t)$ and $AMDE(t)$ for all N robots and for all topologies. This means that the final self-organized aggregations are achieved where all the robots had nearly the same density with the desired inter-robot distance. We also note that the performance of the proposed strategy improved when the size of the swarm increase. However, the overall time convergence of the metrics in absence of obstacles is much better when compared to the obstacle case studies.

7.3. The effect of sensory noise

In this section, we investigate how the aggregation performance was affected when the readings of the robots range and bearing sensors were corrupted by noise. We modelled noise as in the normal experimental simulation, i.e., as a Gaussian distribution with zero mean and a standard deviation $\sigma = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, but this time we considered different values of σ. The other experimental settings (given in Table 1) remained fixed. We set an experiment for each value of σ and performed 25 runs of each experiment. In all of the simulations, we fixed $N = 100$ robots, and we used the following DW-KNN topologies in each run: DW-2NN, DW-3NN, DW-4NN, and DW-5NN.

Fig. 9 plots $F_w(t)$ (Fig. 9a), $F_{ag}(t)$ (Fig. 9b), and $AMDE(t)$ (Fig. 9c) with respect to the different values of σ. In all of the plots, each box represents the metric values obtained from 25 simulations with different initial configurations of 100 robots. The red squares show the mean values, and the dashed red line shows a least squares regression fit to the five points of each topology. The olive-coloured band

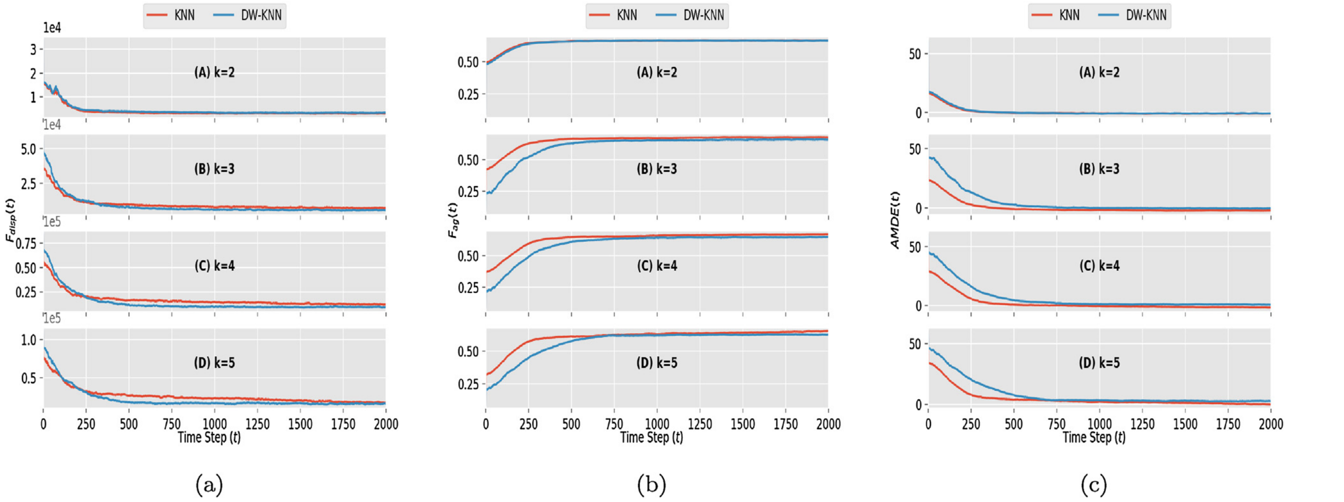


Fig. 10. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the Mean Shift noise model (case $b = 3$): (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

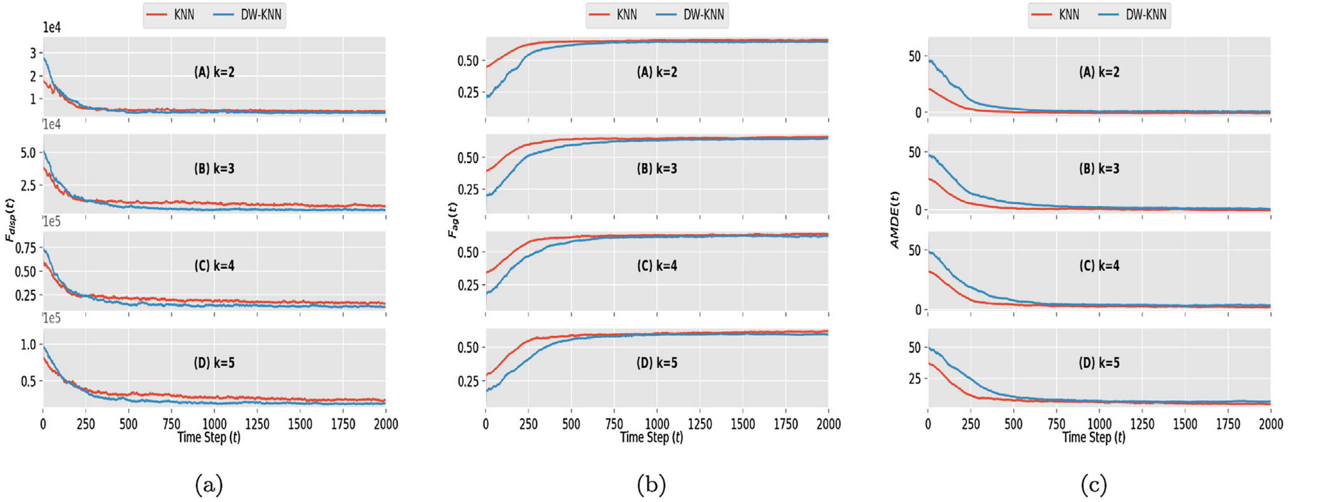


Fig. 11. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the Mean Shift noise model (case $b = 6$): (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

indicates the evolution of the metric with regards to the DW-KNN topology; it represents another linear least squares regression fit to the points of the mean values of each five red squares. All of the least squares regression fitting functions have the form $f(x) = c_1 x + c_0$. The coefficients generated for these functions are highlighted in the tables beside each plot in Fig. 9.

We can see in Fig. 9a that, for a given DW-KNN topology, an increase in σ decreases $F_W(t)$. Table 9a illustrates the corresponding least squares regression functions generated for each topology. It suggests that for a given topology, $F_W(t)$ decreases slightly and sublinearly in function of σ. Moreover, the olive band in Fig. 9a shows that $F_W(t)$ is sublinearly increasing with regard to the number K in the DW-KNN topology.

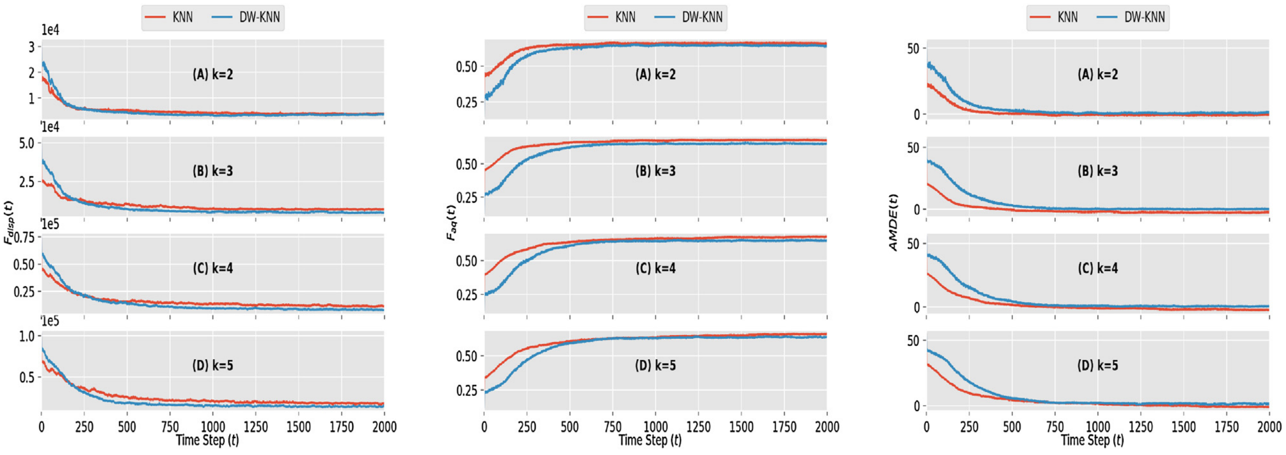
In Fig. 9b, the aggregation quality $F_{ag}(t)$ in a given topology is sublinearly affected by an increase in σ. We can see that an increasing σ yields a decreasing $F_{ag}(t)$. The related least squares regression functions are depicted in Table 9b. However, the olive band in Fig. 9b indicates that $F_{ag}(t)$ shows a small, decreasing deviation with regard to the number K in the DW-KNN topology. Therefore, it remains almost stable.

The metric $AMDE(t)$ was also affected by noise. Fig. 9c shows that as more noise was introduced into the RAB, $AMDE(t)$ deviated further from zero. In each topology, the red dashed lines that represent linear least square regression fitting functions indicates that $AMDE(t)$ sublinearly increases with regards to the value of σ. On the other hand, an analysis of the olive band in Fig. 9c shows that, whatever the value of K is, $AMDE(t)$ has a constant slight, sublinear increase that seems stable.

These observations are as expected. More noise in the RAB yields more mismeasurements of the distances and bearings to neighbouring robots, which immediately affects the quality of all the metrics. While attracting more robots by increasing the number K in the DW-KNN topology does not seems to have a great impact on the quality of the aggregation or the average mean distance error, it does increase the quality of $F_W(t)$.

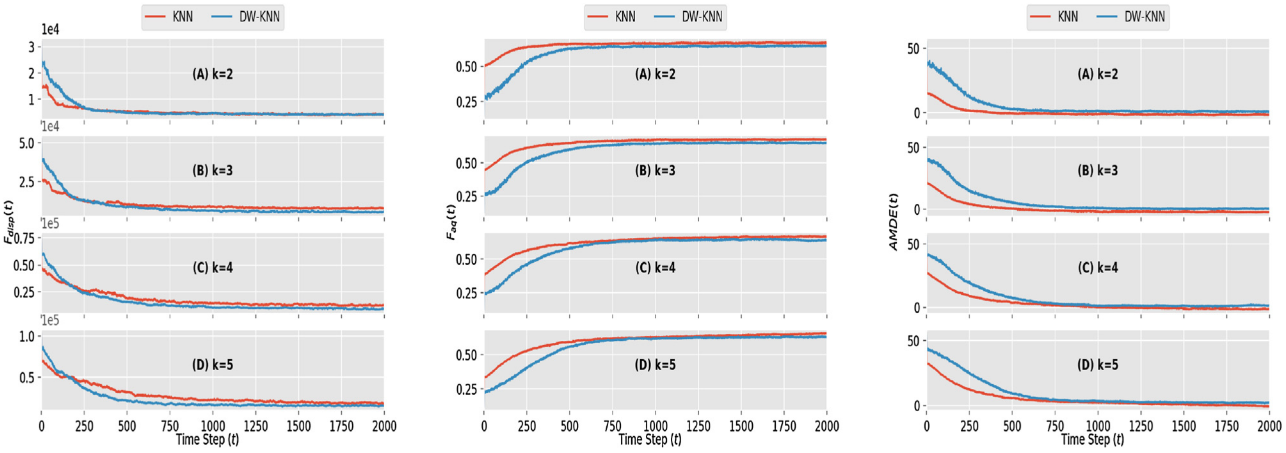
8. Comparison to the KNN aggregation approach

In our previous study (Khaldi and Cherif, 2016a), we proposed a KNN approach for self-organized aggregation in swarm robotics



(a) (b) (c)

Fig. 12. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the Mean Shift noise model (case $b = 6$ and variance 0.1): (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.



(a) (b) (c)

Fig. 13. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the Auto-Correlated noise model where $a = 0.5$: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

systems. In that approach, the interaction between the neighbours was governed by the K-Nearest Neighbours. In this section, we compare the proposed DW-KNN aggregation approach with the KNN one. To better compare the performance of the two approaches, we used another metric called a dispersion metric. This metric has been used by Graham and Sloane (1990) to address the problem of penny packing in a two-dimensional plane. Also, this metric was adopted by Gauci et al. (2014b) to study aggregation task in a swarm of e-puck robots with the assumption of using minimal resources. Unlike the aggregation task that seeks to gather the robots together in an area of an environment, the dispersion metric is generally used to measure the dispersion of a swarm robots-system in the environment. Hence as stated in the study of Graham and Sloane (1990), Gauci et al. (2014b), lower bounds of the metric are analyzed as a bad dispersion quality, meaning that the pennies, as well as the robots, are most close to their centroid, and therefore this can be considered as a good aggregation sign. Here, we follow the same idea and we adopted particularly the dispersion metric proposed by Gauci et al. (2014b) to measure the dispersion of our robots. We call the metric $F_{disp}(t)$, and we used it to measure the quality of dispersion of the entire swarm.

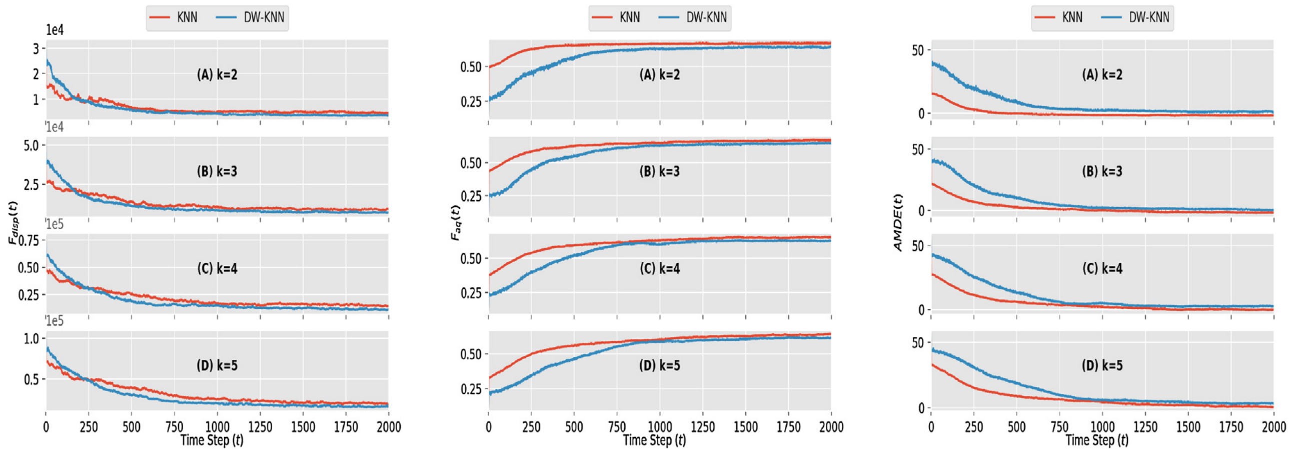
To define this metric, first the dispersion quality $Disp_i(t)$ of a given robot is averaged over its different K neighbours as follows (Gauci et al., 2014b):

$$Disp_i(t) = \frac{1}{4.{r_s}^2}\sum_{i=1}^{K} c_i(t)^2. \qquad (15)$$

Then the dispersion quality, $F_{disp}(t)$, of the swarm is averaged over the number of the robots,

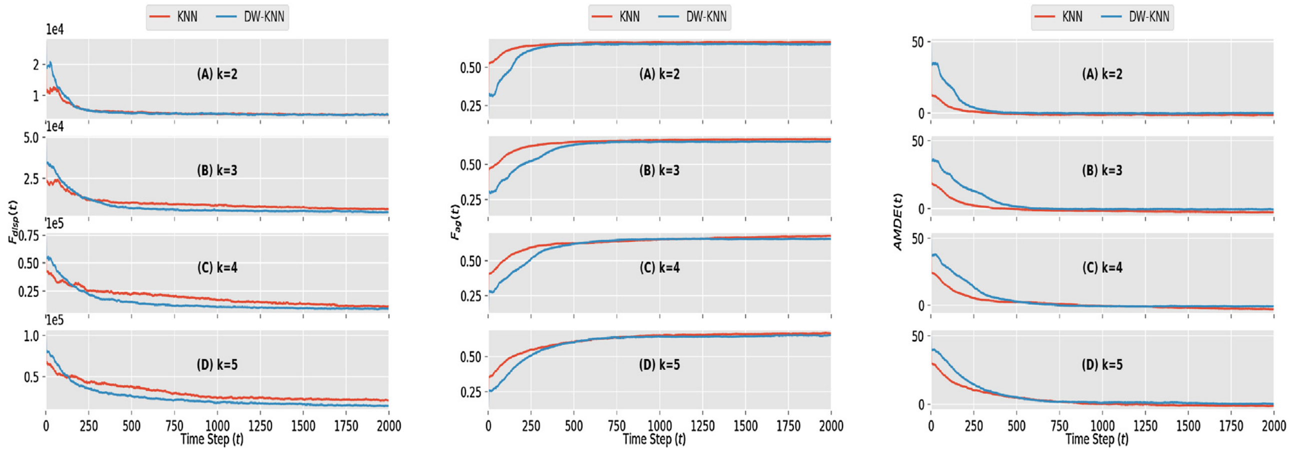
$$F_{disp}(t) = \frac{1}{N}\sum_{i=1}^{N} Disp_i(t). \qquad (16)$$

Where r_s represents the radius of the robot and $c_i(t)$ represent the distance of the robot from the center of mass of the group, and it is computed as in Eq. (11). The $4{r_s}^2$ in the denominator serves to normalize $F_{disp}(t)$ such that it becomes independent of r_s for geometrically similar configurations.



(a) (b) (c)

Fig. 14. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the Auto-Correlated noise model where $a = 0.75$: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.



(a) (b) (c)

Fig. 15. Performance metrics results for the KNN and the DW-KNN topological aggregation approaches with the uniform noise model: (a) $F_{disp}(t)$, (b) $F_{ag}(t)$ and (c) $AMDE(t)$ obtained from the overall controller implemented on $N = 100$ foot-bot robots when taking into consideration different KNN and DW-KNN topologies. In all plots, the x-axis represents the evolution of time step t. Each curve represents the median values obtained from 5 runs with different initial configurations of robots.

Now, we evaluate the performance of the proposed approach as well as the KNN algorithm under the presence of complex noise models. Specifically, the performances of conventional KNN approach and its extension, DW-KNN aggregation method, are compared under different noise models.

8.1. Effect of Gaussian noise with mean shift on the performance of the proposed approach

In this subsection, we investigate the impact of Gaussian noise with a constant bias on the performance of the proposed approach. Towards this end, we introduced a Gaussian noise with a mean shift to both the range and bearing measurements as:

$$Z'_i(t) = Z_i(t) + \varepsilon_i(t), \tag{17}$$

where $Z'_i(t)$ is the noisy measurement, $Z_i(t)$ is the true measurement, and $\varepsilon_i(t)$ is Gaussian noise with constant mean b and unit variance. Using this noise model, we evaluated the performance of both the DW-KNN and the KNN aggregation approaches under the presence of measurement noise with two different levels (i.e., $b = 3$, 6). Fig. 10 shows the results of the quality metrics when a constant bias of amplitude equal $b = 3$, whereas Fig. 11 plots the results when $b = 6$. From Fig. 10, it can be seen that both approaches converge to the same $F_{ag}(t)$ and $AMDE(t)$ values (see Fig. 10b and c for the $b = 3$ case, and Fig. 11b and c for the $b = 6$ case). However, the DW-KNN has a slightly lower dispersion compared to that of the KNN method (see Fig. 10a and Fig. 11a). Lower dispersion means that the robots are very close to their group center and therefore the overall aggregation performance of the DW-KNN is much better than the KNN aggregation method.

In another simulation setup, we conducted simulation with a mean shift constant $b = 0.05$ and a variance 0.1. Results are shown in Fig. 12. Similar to the above noise models, the KNN and the DW-KNN aggregation approaches converge almost to the same quality of aggregation, $F_{ag}(t)$ and to the same mean distance error $AMDE(t)$ (see Fig. 12b and c). However, the overall aggregation performance of the DW-KNN approach is better than the KNN due to the fact that it gives lower dispersion quality than the KNN approach in the case of $K = \{3, 4, 5\}$ (see Fig. 12a). Notice also that for both approaches in the above cases studies, $F_{disp}(t)$ converges almost to the same value in case $k = 2$ and that it slightly increases in case $k = \{3, 4, 5\}$. This is as expected, aggregating robots by forming a mesh of more than two robots ($k > 2$), will attract more weighted robots to the center of their group when using a DW-KNN approach than the KNN approach. It can also be seen that when noise level increases the aggregation quality in term of the dispersion degrade by increasing the dispersion metric. The results also show that moderate noise level has no significant impact on the $AMDE(t)$ and $F_{ag}(t)$.

8.2. Effect of autocorrelated noise on the performance proposed approach

In this subsection, the performance of the DW-KNN and KNN-based neighborhood topology will be investigated in the presence of autocorrelated noise. Towards this end, the performance of DW-KNN and KNN approaches have been studied when the measurement noise is generated from a first-order autoregressive process, or AR(1). Specifically, the measurement noise of the range and bearing sensors are generated using an AR(1) model as follow:

$$\begin{cases} Z_i'(t) = Z_i(t) + \zeta_i(t) \\ \zeta_i(t) = a\zeta_i(t-1) + \varepsilon_i(t) \end{cases}, \tag{18}$$

where $Z_i'(t)$ is the noisy measure, $Z_i(t)$ is the true measure, a is the autocorrelation coefficient with lag 1, and $\epsilon_i(t)$ is a Gaussian distribution of 0 means and a standard deviation 1.

With this noise model, we investigated the performance of the two approaches by conducting simulations for different values of the AR-parameter $a = \{0.5, 0.75\}$. Figs. 13 and 14 plot successively the results of the two based topological approaches for the different values of a. In both values of a, the analysis results are close to the analysis depicted in the previous noise models where the two approaches converge nearly to the same $F_{ag}(t)$ and the same $AMDE(t)$ in a hand. In another hand, the DW-KNN approach gives more dispersion quality than the KNN approach (a lower value is better). In addition, the two approaches converge almost to the same dispersion quality in case $k = 2$ and that they slightly diverge linearly to different values in case $k = \{3, 4, 5\}$. This is also as expected when grouping the robots by forming a mesh of more than two robots ($k > 2$), using a DW-KNN approach will attract more weighted robots to the center of their group than the KNN approach. Alike the previous noise model, it can be also seen that when AR coefficient increases (highly correlated noise) the aggregation quality in term of the dispersion degrade by increasing the dispersion metric, and the results also show that noise with moderate autocorrelation has no significant impact on the $AMDE(t)$ and $F_{ag}(t)$. Notice also that the overall aggregation metrics take much time steps to be converged while compared to the previous noise models. As can be seen in all the related figures, the time convergence of the metrics is proportional to the value of the autoregressive coefficient (for a greater value of a the time convergence of the metrics is long).

8.3. Effect of uniform noise on the performance proposed approach

In this case study, range and bearing sensors noise are generated from a uniform distribution of [−3, 3 cm] for a range measurement and of [−5°, 5°] for a bearing measure. Fig. 15 illustrates the results of KNN and DW-KNN based topological approaches. Like the other noise models, both the KNN and the DW-KNN aggregation approaches converge almost to the same quality of aggregation and to the same mean distance error (see Fig. 15b and c). But the DW-KNN approach gives lower dispersion quality than the KNN approach in the case of $K = \{3, 4, 5\}$ (see Fig. 15c), which mean that the robots are better close to their center of masse, and hence it is a good sign of aggregation performance. However, the two approaches perform slightly in a similar way when $K = 2$. This can be expected as in the previous noise models, when grouping robots by more then two robots ($k > 2$), the DW-KNN approach will attract more weighted robots to the center of their group than the KNN one. Also as analyzed in the previous noise models, the aggregation quality in term of the dispersion degrade by increasing the dispersion metric, and the results also show that uniform noise model has no significant impact on the $AMDE(t)$ and $F_{ag}(t)$.

9. Conclusions and further work

When studying the collective behaviours of a large number of individual robots, the inter-robot distance can be of key importance, but additional properties such as the density of robots could have a greater impact on the collective behaviour of the whole swarm. Here we relied on both the distance and the density as equal key factors, and we proposed a DW-KNN-based neighbourhood topology to study self-organization in an aggregating robot swarm. With this topology, we used the virtual viscoelastic-based proximal model developed in our previous work (Khaldi and Cherif, 2016a) to keep and arrange the neighbours together. DW-KNN was achieved by defining a distance-weighted function based on an SPH interpolation approach to estimate the density of robots in the swarm.

Also, the proposed algorithm exhibited high efficiency to smoothly drive the robots swarm to achieve cubic based self-organized aggregations in the presence and absence of obstacles in the arena. The proposed approach could be helpful in a situation when attracting a large scale of robots from one area to another while maintaining a connectedness between the robots and avoiding collisions.

Various self-organized aggregations are achieved using the ARGoS simulator in both absence and presence of obstacles, and performance analysis within three metrics shows the efficacy of the proposed approach. The effect of noise in the robot range and bearing sensing capabilities is also addressed in this study showing how the proposed model is behaving in such circumstance.

Furthermore, the proposed method showed superior performance compared to the KNN approach in presence of different noise models (i.e., uniform, Gaussian noise with mean-shift, and Auto-Correlated noise models).

Moreover, in the present approach the choice of the density estimation method to be used to weight the distances could play a crucial role in the emergence of the self-organized aggregations. This is let to the scenario in which the approach will be applied. For example, we believe that with the proposed aggregation approach, various B-spline based self-organized aggregations could easily emerge by adopting other kernel functions in the SPH density estimation methods such as those based on the Schoenberg B-spline functions (Schoenberg, 1988) (i.e., the M5 quartic functions and the M6 quantic functions (Price, 2012)). Moreover, different density estimation methods that differ from the SPH approach (i.e., Gaussian kernel density estimation method) could be used as an alternative to achieve different self-organized aggregations, or to drive the swarm of robots to a desired density distribution. Therefore, we open a door for further studies on how to select the density estimation method for a given scenario.

In the light of the above points and as future work, we plan to filter measurements noise to enhance the performance of the model. We plan also to study the impact of the robot density on self-organized aggregation by incorporating different SPH density-estimation methods and using other density estimation technics that differs from the SPH one. More attention should also be given to how a robot swarm collectively decides to adapt the values of K to dynamically switch its neighbouring relationships.

Acknowledgements

This publication is based upon a collaboration work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2015-CRG4-2582, and the LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, Biskra, Algeria. We would like to thank the reviewers of this article for their insightful comments, which helped us to greatly improve its quality.

Appendix A. The overall control algorithm implemented in a foot-bot robot

Algorithm 1. Part I

```
Global Parameteres: $Define\ the\ global\ parameters\ in\ Table.1$
struct {
    float: $distance, angle, speed, weight$
} $Vect$
Function init()
    Initialize Global Parameters
    return
// function executed every time step
Function step()
    // Send $v_i$ to neighbor via RAB
    $robot.range\_and\_bearing.set\_data(1, v_i)$
    // Call the SPH Density Estimation Control Function
    $\rho_i \longleftarrow$ SPHDEC()
    // Send $\rho_i$ to neighbor via RAB
    $robot.range\_and\_bearing.set\_data(2, \rho_i)$
    // Build the Distance-Weighted KNN mesh
    $\mathcal{T}_i \longleftarrow$ DWKNNC()
    // Call the Proximal Control Function
    $\hat{p}_i(x, y) \longleftarrow$ PC($\mathcal{T}_i$)
    // Call the Repulsive Control Function
    $\hat{r}_i(x, y) \longleftarrow$ RC()
    // Compute vector $\hat{a}_i$
    $\hat{a}_i(x, y) \longleftarrow (\hat{p}_i.x + \hat{r}_i.x, \hat{p}_i.y + \hat{r}_i.y)$
    // Call the Motion Control Function
    $(v_{l_i}, v_{r_i}) \longleftarrow$ MC($\hat{a}_i$)
    // actuate the robot' s wheel speeds
    robot.wheels.set_velocity $(v_{l_i}, v_{r_i})$
    return
Function SPHDEC()
    Using: $Range\ and\ Bearing\ Sensors\ (RAB)$
    Input: $h$
    $\rho_i \longleftarrow 0$
    // Collect data from the robot local RAB sensors
    for $j = 1..|robot.range\_and\_bearing|$ do
        $d_{ij} \longleftarrow robot.range\_and\_bearing[i].range$
        // Compute $W(d_{ij}, h)$ and update $\rho_i$ using eq.(4)
        $W \longleftarrow computeW(d_{ij}, h)$
        $\rho_i \longleftarrow \rho_i + W$
    return $\rho_i$
```

Algorithm 2. Part II

```
Function PC()
    Input: $v_i, \mathcal{T}_i$
    $result(x, y) \longleftarrow (0, 0)$;
    // Collect data from the robot local RAB sensors
    for $j = 1..|\mathcal{T}_i|$ do
        // compute virtual voigt force generated by neighbor j using equation (2)
        $f(x, y) \longleftarrow ComputeVoigtForce(\mathcal{T}_i[j], v_i)$;
        $result(x, y) \longleftarrow result(result.x + f.x, result.y + f.y)$;
    return $result(x, y)$;
Function RC()
    Using: $Proximity\ Sensors$
    $result(x, y) \longleftarrow (0, 0)$;
    $\mathcal{F}_i(x, y) \longleftarrow (0, 0)$;
    for $i = 1..24$ do
        $x_i \longleftarrow robot.proximity[i].value * cos(robot.proximity[i].angle)$;
        $y_i \longleftarrow robot.proximity[i].value * sin(robot.proximity[i].angle)$;
        $\mathcal{F}_i(x, y) \longleftarrow \mathcal{F}_i(\mathcal{F}_i.x + x_i, \mathcal{F}_i.y + y_i)$;
    // Now $\mathcal{F}_i$ is the vector that points to the direction to the closest obstacle
       where $\parallel \mathcal{F}_i \parallel = \sqrt{\mathcal{F}_i.x^2 + \mathcal{F}_i.y^2}/24$ and $\angle\mathcal{F}_i = atan2(\mathcal{F}_i.y, \mathcal{F}_i.x)$
    // Compute the repulsive force vector using eq.6
    $result(x, y) \longleftarrow computeRepulsiveForce(\mathcal{F}_i(x, y))$
    return $result(x, y)$;
Function DWKNNC()
    Using: $Range\ and\ Bearing\ Sensors\ (RAB)$
    Input: $K$
    $\mathcal{S} \longleftarrow \emptyset$;
    // Collect data from the robot RAB sensors
    for $i = 1..|robot.range\_and\_bearing|$ do
        $\mathcal{V} \longleftarrow new(Vect)$;
        $\mathcal{V}.distance \longleftarrow robot.range\_and\_bearing[i].range$;
        $\mathcal{V}.angle \longleftarrow robot.range\_and\_bearing[i].horizontal\_bearing$;
        $\mathcal{V}.speed \longleftarrow robot.range\_and\_bearing[i].data[1]$;
        $\mathcal{V}.weight \longleftarrow robot.range\_and\_bearing[i].data[2] * robot.range\_and\_bearing[i].range$;
        $\mathcal{S} \longleftarrow \mathcal{S} \cup \mathcal{V}$;
    $SortByWeight(\mathcal{S})$;
    $\mathcal{R} \longleftarrow getTheFirstKElemnts(\mathcal{S}, K)$;
    return $\mathcal{R}$;
Function MC(force)
    Input: $force$
    // Compute the angular and the forward speed $(\omega_i, v_i)$ using equation (8)
    $(\omega_i, v_i) \longleftarrow computeAngularAndForwardSpeed(force)$;
    // Compute the velocities $(v_{li}, v_{ri})$ of the left and the right wheels of the robot
       using equation (7)
    $(v_{li}, v_{ri}) \longleftarrow computeLeftAndRightSpeed(\omega_i, v_i)$;
    return $(v_{li}, v_{ri})$;
```

References

Ando, H., Oasa, Y., Suzuki, I., Yamashita, M., 1999. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Trans. Robot. Autom. 15 (5), 818–828.

Arvin, F., Samsudin, K., Ramli, A.R., Bekravi, M., 2011. Imitation of honeybee aggregation with collective behavior of swarm robots. Int. J. Comput. Intell. Syst. 4 (4), 739–748.

Arvin, F., Turgut, A.E., Bellotto, N., Yue, S., 2014a. Comparison of different cue-based swarm aggregation strategies. In: International Conference in Swarm Intelligence. Springer, pp. 1–8.

Arvin, F., Turgut, A.E., Bazyari, F., Arikan, K.B., Bellotto, N., Yue, S., 2014b. Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. Adapt. Behav. 22 (3), 189–206.

Arvin, F., Turgut, A.E., Krajník, T., Yue, S., 2016. Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm. Adapt. Behav. 24 (2), 102–118.

Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., et al., 2008a. Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. Proc. Natl. Acad. Sci. 105 (4), 1232–1237.

Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., et al., 2008b. Empirical investigation of starling flocks: a benchmark study in collective animal behaviour. Anim. Behav. 76 (1), 201–215.

Bodi, M., Thenius, R., Szopek, M., Schmickl, T., Crailsheim, K., 2012. Interaction of robot swarms using the honeybee-inspired control algorithm beeclust. Math. Comput. Model. Dyn. Syst. 18 (1), 87–100.

Bodi, M., Möslinger, C., Thenius, R., Schmickl, T., 2015. BEECLUST used for exploration tasks in autonomous underwater vehicles. In: 8th Vienna International Conference on Mathematical Modelling, vol. 48, no. 1, IFAC, pp. 819–824.

Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., Mondada, F., 2010. The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 4187–4193.

Broly, P., Mullier, R., Deneubourg, J.-L., Devigne, C., 2012. Aggregation in woodlice: social interaction and density effects. ZooKeys 176, 133.

Broly, P., Deneubourg, J.-L., Devigne, C., 2013. Benefits of aggregation in woodlice: a factor in the terrestrialization process? Insectes Sociaux 60 (4), 419–435.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., Bonabeau, E., 2002. Self-Organization in Biological Systems. Princeton University Press.

Correll, N., Martinoli, A., 2011. Modeling and designing self-organized aggregation in a swarm of miniature robots. Int. J. Robot. Res. 30 (5), 615–626.

De Gennaro, M.C., Jadbabaie, A., 2006. Decentralized control of connectivity for multi-agent systems. In: 2006 45th IEEE Conference on Decision and Control, IEEE, pp. 3628–3633.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., et al., 2004. Evolving self-organizing behaviors for a swarm-bot. Auton. Robots 17 (2), 223–245.

Ercan, M.F., Li, X., Liang, X., 2010. A regular tetrahedron formation strategy for swarm robots in three-dimensional environment. In: International Conference on Hybrid Artificial Intelligence Systems. Springer, pp. 24–31.

Garnier, S., Jost, C., Gautrais, J., Asadpour, M., Caprari, G., Jeanson, R., Grimal, A., Theraulaz, G., 2008. The embodiment of cockroach aggregation behavior in a group of micro-robots. Artif. Life 14 (4), 387–408.

Gasparri, A., Priolo, A., Ulivi, G., 2012a. A swarm aggregation algorithm for multi-robot systems based on local interaction. In: 2012 IEEE International Conference on Control Applications (CCA), IEEE, pp. 1497–1502.

Gasparri, A., Oriolo, G., Priolo, A., Ulivi, G., 2012b. A swarm aggregation algorithm based on local interaction for multi-robot systems with actuator saturations. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 539–544.

Gauci, M., Chen, J., Dodd, T.J., Groß, R., 2014a. Evolving aggregation behaviors in multi-robot systems with binary sensors. In: Distributed Autonomous Robotic Systems. Springer, pp. 355–367.

Gauci, M., Chen, J., Li, W., Dodd, T.J., Groß, R., 2014b. Self-organized aggregation without computation. Int. J. Robot. Res. 33 (8), 1145–1161.

Gordon, N., Wagner, I.A., Bruckstein, A.M., 2004. Gathering multiple robotic a(ge)nts with limited sensing capabilities. In: International Workshop on Ant Colony Optimization and Swarm Intelligence. Springer, pp. 142–153.

Gordon, N., Elor, Y., Bruckstein, A.M., 2008. Gathering multiple robotic agents with crude distance sensing capabilities. In: International Conference on Ant Colony Optimization and Swarm Intelligence. Springer, pp. 72–83.

Gou, J., Du, L., Zhang, Y., Xiong, T., et al., 2012. A new distance-weighted k-nearest neighbor classifier. J. Inf. Comput. Sci. 9 (6), 1429–1436.

Graham, R.L., Sloane, N.J., 1990. Penny-packing and two-dimensional codes. Discrete Comput. Geometry 5 (1), 1–11.

Hemelrijk, C.K., Hildenbrandt, H., 2012. Schools of fish and flocks of birds: their shape and internal structure by self-organization. Interface Focus, rsfs20120025.

Holland, O., Melhuish, C., 1997. An interactive method for controlling group size in multiple mobile robot systems. In: Proceedings, 8th International Conference on Advanced Robotics, 1997. ICAR'97, IEEE, pp. 201–206.

Jeanson, R., Rivault, C., Deneubourg, J.-L., Blanco, S., Fournier, R., Jost, C., Theraulaz, G., 2005. Self-organized aggregation in cockroaches. Anim. Behav. 69 (1), 169–180.

Kernbach, S., Thenius, R., Kernbach, O., Schmickl, T., 2009. Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. Adapt. Behav. 17 (3), 237–259.

Khaldi, B., Cherif, F., 2016a. A virtual viscoelastic based aggregation model for self-organization of swarm robots system. In: Conference Towards Autonomous Robotic Systems. Springer, pp. 202–213.

Khaldi, B., Cherif, F., 2016b. Swarm robots circle formation via a virtual viscoelastic control model. In: 2016 8th International Conference on Modelling, Identification and Control (ICMIC), IEEE, pp. 725–730.

Khaldi, B., Harrou, F., Cherif, F., Sun, Y., 2017. Monitoring a robot swarm using a data-driven fault detection approach. Robot. Auton. Syst. 97, 193–203.

Kube, C.R., Zhang, H., 1993. Collective robotics: from social insects to robots. Adapt. Behav. 2 (2), 189–218.

Leccese, A., Gasparri, A., Priolo, A., Oriolo, G., Ulivi, G., 2013. A swarm aggregation algorithm based on local interaction with actuator saturations and integrated obstacle avoidance. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1865–1870.

Lee, G., Chong, N.Y., 2008. Flocking Controls for Swarms of Mobile Robots Inspired by Fish Schools. INTECH Open Access Publisher.

Mermoud, G., Matthey, L., Evans, W.C., Martinoli, A., 2010. Aggregation-mediated collective perception and action in a group of miniature robots. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, International Foundation for Autonomous Agents and Multiagent Systems, pp. 599–606.

Merris, R., 1994. Laplacian matrices of graphs: a survey. Linear Algebra Appl. 197, 143–176.

Niizato, T., Murakami, H., Gunji, Y.-P., 2014. Emergence of the scale-invariant proportion in a flock from the metric-topological interaction. Biosystems 119, 62–68.

Pimenta, L.C., Michael, N., Mesquita, R.C., Pereira, G.A., Kumar, V., 2008. Control of swarms based on hydrodynamic models. In: IEEE International Conference on Robotics and Automation, 2008. ICRA 2008, IEEE, pp. 1948–1953.

Pimenta, L.C., Pereira, G.A., Michael, N., Mesquita, R.C., Bosque, M.M., Chaimowicz, L., Kumar, V., 2013. Swarm coordination based on smoothed particle hydrodynamics technique. IEEE Trans. Robot. 29 (2), 383–399.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., et al., 2012. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intell. 6 (4), 271–295.

Pini, G., Dorigo, M., Birattari, M., 2009. Towards Collective Robotics in a 3D Space: Simulation With Hand-Bot Robots, IRIDIA. Université Libre de Bruxelles, Tech. Rep.

Price, D.J., 2012. Smoothed particle hydrodynamics and magnetohydrodynamics. J. Comput. Phys. 231 (3), 759–794.

Schoenberg, I.J., 1988. Contributions to the problem of approximation of equidistant data by analytic functions. In: IJ Schoenberg Selected Papers. Springer, pp. 3–57.

Soysal, O., Sahin, E., 2005. Probabilistic aggregation strategies in swarm robotic systems. In: Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE, pp. 325–332.

Soysal, O., Bahçeci, E., ŞahİN, E., 2007. Aggregation in swarm robotic systems: evolution and probabilistic control. Turk. J. Electr. Eng. Comput. Sci. 15 (2), 199–225.

Spears, W.M., Spears, D.F., Hamann, J.C., Heil, R., 2004. Distributed, physics-based control of swarms of vehicles. Auton. Robots 17 (2–3), 137–162.

Trianni, V., Groß, R., Labella, T.H., Şahin, E., Dorigo, M., 2003. Evolving aggregation behaviors in a swarm of robots. In: European Conference on Artificial Life. Springer, pp. 865–874.

Violeau, D., Rogers, B.D., 2016. Smoothed particle hydrodynamics (sph) for free-surface flows: past, present and future. J. Hydraul. Res. 54 (1), 1–26.

Wahby, M., Weinhold, A., Hamann, H., 2016. Revisiting beeclust: aggregation of swarm robots with adaptiveness to different light settings. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 272–279.

Zhao, S., Ramakrishnan, S., Kumar, M., 2011. Density-based control of multiple robots. In: Proceedings of the 2011 American Control Conference, IEEE, pp. 481–486.