Flexible and Efficient Topological Approaches for a Reliable Robots Swarm Aggregation

BELKACEM KHALDI¹, FOUZI HARROU², (MEMBER, IEEE), FOUDIL CHERIF¹ AND YING SUN².

¹LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, R.P. 07000 Biskra, Algeria.
²King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, Thuwal, 23955-6900, Saudi Arabia.

Corresponding authors: Fouzi Harrou (e-mail: fouzi.harrou@kaust.edu.sa)

The authors (Belkacem Khaldi and Foudil Cherif) would like to thank the LESIA Laboratory, Department of Computer Science, University of Mohamed Khider, Biskra, Algeria for the continued support during the research. This work is supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7-3800.

ABSTRACT Aggregation is a vital behavior when performing complex tasks in most of the swarm systems such as swarm robotics systems. In this paper, three new aggregation methods, namely the Distance-Angular, the Distance-Cosine, and the Distance-Minkowski $k$-nearest neighbor ($k$-NN) have been introduced. These aggregation methods are mainly built on well-known metrics: the Cosine, Angular and Minkowski distance functions, which are used here to compute distances among robots neighbors. Relying on these methods, each robot identifies its $k$ nearest neighborhood set that will interact with. Then in order to achieve the aggregation, the interactions sensing capabilities among the set members are modeled using a virtual viscoelastic mesh. Analysis of the results obtained from the ARGoS simulator shows a significant improvement in the swarm aggregation performance while compared to the conventional distance-weighted $k$-NN aggregation method. Also, the aggregation performance of the methods is reported to be robust to partially faulty robots and accurate under noisy sensors.

INDEX TERMS Swarm Robotics, Self-organized aggregation, $k$-NN, Virtual viscoelastic mesh, Distance metrics, Partially faulty robots.

I. INTRODUCTION

LIVING in a group of similar individuals is a fundamental behavior that is often observed in many biological organisms such as flocks of birds, schools of fish and mammals of herds [1], [2]. In fact, this behavior is simply called aggregation and plays a crucial role in performing complex tasks that are beyond the capabilities of a single individual. The reasons for living in aggregations may differ from one species to another. For example, the anti-predator reaction is one of the traditional motivations behind staying together for most group-living animals [3]. Murmuration of a large number of flocks of birds, such as starlings, is a mesmerizing phenomenon that is consistently obtained through aggregation [4]. Other abilities, guaranteed by remaining together, can be observed in other biological collective behaviors such as the construction of a nest by termites [5] or the formation of a spore-bearing structure by slime mold [6].

From biologists’ point of view, these aggregation behaviors are commonly achieved in a self-organized manner [1], [3]. The particularity of this type of aggregation is that it can be achieved in random areas without any external stimulus and in the total absence of a central orchestrator [7]. Moreover, it is regularly achieved in a cohesive manner using only simple and local interaction rules among individuals.

On the basis of such biological studies, the self-organized aggregation behavior has also been addressed in various swarm robotics studies [8]–[10]. It’s a fundamental requirement for robots swarm to perform complex swarming behaviors, such as self-assembly, pattern formation, and coordinated movement or for information exchanging [11]–[13].

In a previous work [14], the authors were inspired by the topological distance metric revealed in the collective behavior studies of bird flocks and fish schools [15], [16]. It is reported in those studies that the agents interact only with
the six or seven neighbors available in each agent’s field of view. While few works have applied such topological metrics to the study of swarm robotics collective behaviors [17]–[19]. However, most of these studies rely on distances as the only constraint. Recently in [20], [21], the authors proposed an approach to study swarm robots self-organized aggregation using a topological interaction strategy according to two main constraints: distance and density. The interacting strategy was governed by a Distance-Weighted K-Nearest Neighboring (DW-KNN) technique, in which the robots are virtually attracted to the nearest \( k \) neighbors with the smallest weighted distances. Here, the distances to the neighbors are weighted using a density-based estimation technique, then each robot had to select the DW-KNNs with which a virtual viscoelastic mesh is built. Results within this approach, under different noise models on the robots’ sensors, showed efficient performance when compared to the KNN approach.

In this work, we seek to improve the aggregation performance of the DW-KNN method by applying machine learning \( k \)-NN based distance metrics. We specifically aim to investigate how such distance functions can be used to compute distances among robots’ neighbors and to enhance the swarm aggregation performance. To the best of the authors’ knowledge, such metrics have not yet been applied to study collective behaviors within swarm robotics systems as well as multi-agent systems.

In machine learning, the \( k \)-NN algorithm is a simple technique used for predicting the output of a new instance according to the majority class/value of its \( k \) closest training instances [22]. Finding the \( k \) nearest training instances can be done on the basis of similarities between instances by calculating specific distance metrics such as the Euclidean metric. Depending on the properties of data, there are other distance metrics (e.g. Cosine, Angular, Minkowski, . . .) that can be applied in the literature [22]–[24].

Inspired by such distance functions, we have so far proposed three newly swarm robots self-organized aggregation control models (Section IV), namely: the Distance-Angular \( k \)-NN (DA-KNN), the Distance-Cosine \( k \)-NN (DC-KNN), and the Distance-Minkowski \( k \)-NN (DM-KNN). These control models are built on well-known metrics: the cosine, angular and Minkowski distance functions (Section IV-A), which are used here to calculate the distances between neighboring robots. Based on the proposed methods, the nearest \( k \) neighborhood group of each robot with which a robot will interact is identified (Section IV-B). Then, to achieve aggregation, the interaction-sensing capabilities among the members of the group are modeled using a virtual viscoelastic mesh. The overall aggregation control models for the proposed methods are implemented on a swarm of foot-bot robots simulated using the ARGoS simulator [25]. Experimental results with analysis of the minimum bounding rectangle area that encloses the entire robots swarm are reported in Section V. The results show comparative illustrations between the proposed aggregation methods and the previous DW-KNN method. Furthermore, to more precisely evaluate the aggregation performances of the methods, we analyze them using the swarm dispersion metric [21], [26] (Section VI-A). Analytic studies within that metric involve investigating the performances of the methods under normal circumstances (Section VI-B), in presence of sensory noise (Section VI-C) and presence of abnormal robots (Section VI-D). Finally, we sum up and discuss some of the future perspectives of this work in Section VII. In the following and before we dig into the details of the proposed methods, we shall review in Section II some of the related works and then we formally introduce in Section III the problem statement.

II. RELATED WORKS

The problem of self-organized aggregation has been widely studied in the literature of swarm robotics. So far, different approaches have been suggested to find solutions to such a problem. In the following subsections, we highlight the most famous of them.

A. PROBABILISTIC APPROACH

In this approach, aggregation is achieved by controlling robots via probabilistic finite state machines (PFSMs). Most of the related works were inspired by Jeanson et al. [27]’s cockroach model. In this model, joining or living clusters is reported to be with probabilities that rely on the clusters’ sizes. A high probability is assigned to a cockroach to join a larger cluster and a lower probability for leaving a smaller cluster. Similarly, Garnier et al. [28] succeeded in mimicking this behavior and obtaining efficient aggregation results in homogeneous environments within a swarm of Alice’s physical robots. When analyzing a similar model to the above ones, Correll and Martinoli [29] reported a probabilistic aggregation method requiring at least a combination of locomotion speed and communication range to obtain a single aggregation cluster. In another work, Soysal and Sahin [30] combined a set of simple behaviors rules in a PFSM to achieve a generic aggregation behavior within robotics swarm system. Recently, Nicolas et al. [31] introduced a naming game model [32] to the aggregation model of Garnier et al. [28] and studied the interaction effect between them. With this combination, promising results in extending the naming game and aggregation capabilities were revealed.

B. EVOLUTIONARY APPROACH

In this approach, the aggregation dynamics is achieved using artificial evolution technique. Here, the robot is commonly controlled by a neural network that maps sensory inputs to the robot’s actuators outputs. In this context, a minimalist aggregation model based on an evolutionary method was proposed by Gauci et al. [26], [33] to study self-organized aggregation within a swarm of e-pucks. The authors reported that their model is able to perform emergent aggregation without computation using only a single binary sensor that could detect the existence of another robot in its line of vision. However, experimental and simulation results have shown that the binary sensor should have a range long enough
to perform an accurate aggregation. A neural network model was used by Trianni et al. [34] to control simple robots, called s-bots, to study collective aggregation behavior. Model parameters were evolved using a genetic approach leading, according to the authors, to produce two kinds of clustering behavior: dynamic and static aggregation. This model was then extended by Dorigo et al. [35] who reported that in addition to the aggregation behavior, coordination motion behavior could also be achieved using efficient evolved controllers. Similarly, Soysal et al. [36] introduced a number of features such as the number of robots and the size of the area to evaluate their effect on the performance and scalability of the evolved aggregation behavior. In other studies, Gomes et al. [37], [38] exploited the novelty search concept to evolve neural controllers to study aggregation in robots swarm systems. Results of the simulation, obtained by the studies, showed that a promising alternative to the classical evolution of swarm robotics controllers is guaranteed by the application of the concept of novelty research. In [39], the authors applied a particle swarm optimization (PSO) strategy to guide a swarm robots system divided into groups to search for multiple targets simultaneously. Under this improved strategy, robots are able to be grouped after a number of stochastic movements iterations and they are supposed to find all targets finally.

C. ARTIFICIAL PHYSICS APPROACH

In this approach, the aggregation task is mainly achieved by applying virtual physical laws between the robots. In a work done by Gasparri et al. [40], the interaction between robots is governed by attractive/repulsive virtual force laws, which result in an artificial physics model leading to a successful aggregation performance in a swarm robots system. Later, the authors extended their model to deal with actuator saturation [41], then Leccese et al. [42] extended the model to cope with obstacle avoidance. In another study, Jito and Bibhya [43] studied the collective behavior of moving swarm agents using a Lagrangian-based model. This model is built upon a Lypanuve function that takes into account long-range attraction and short-range repulsion as interaction rules between agents. Computer simulations have shown that in addition to the basic swarm features such as cohesion, stability, and aggregation that the model is able to capture, other complex dynamics similar to those seen in natural social animals could also be exhibited. In a previous study [14], we studied the collective aggregation behavior in a swarm of foot-bots robots using a topological approach in which the inter-robots interactions are modeled via virtual viscoelastic forces. In that work, the foot-bots interact only with the K-nearest neighbors (KNNs) rather than all the neighbors who are in their field of vision. The KNNs were selected by each robot based solely on the distance constraint. Later in [21], a smoothed particle hydrodynamics (SPH) density estimation technique was used with the distance as two constraints to develop a weighted-distance topological approach to study self-organized aggregation within the same robots swarm system. Promising simulation results from the ARGoS simulator [25] showed that the two topological approaches are able to emerge various self-organized aggregating patterns even in presence of obstacles and that the weighted-distance KNN aggregation approach outperforms the classical KNN approach in both absence and existence of obstacles.

III. PROBLEM STATEMENT

Consider a 2-D, bounded rectangular area, surrounded by four walls, with no obstacles or other cues placed in it. Inside the area, a swarm of $N$ mobile agents is initially distributed in random positions and oriented towards arbitrary directions. The agent is able to move using a differential two-wheel-drive configuration, meaning that its motion is driven by the forward speeds of the left and right wheels, which are respectively denoted $v_l$ and $v_r$. Having $b$ as the distance between the wheels of the agent, $v_l$ and $v_r$ can be described according to the driving and angular velocities $v_i$ and $\omega_i$ of the agent as follows [44]:

$$\begin{bmatrix} v_l \\ v_r \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix},$$

(1)

Note that to preserve the limited capacity of the agent’ motor, the absolute value of $v_i$ and $\omega_i$ cannot exceed a maximum speed, $v_{max}$ and $\omega_{max}$, respectively [45].

At each time step, $t$, an agent, $i$, can communicate with its neighbors, $N_i(t)$, that are in its communication range, $Dr$. During a communication phase, each agent is able to locally measure the range, $d_{ij}$, and bearing, $\theta_{ij}$, of the $j$th neighbor, where $j \in N_i(t)$.

The main objective is the design of control inputs that will drive the swarm of $N$ agents to aggregate in somewhere place inside the arena without relying on any external cues. Here, the meaning of somewhere place inside the arena is that the agents have to form aggregations without any predefined aggregation zones. So, the problem to solve now is to find the control parameters leading to the desired aggregation behavior.

IV. AGGREGATION CONTROL MODELS

To approach the above problem, we propose three different aggregation control models: the Distance-Angular based $k$-NN, Distance-Cosine based $k$-NN, and Distance-Minkowski based $k$-NN methods. The foundation of these methods is based on our previous Distance-Weighted $k$-NN control model, in which virtual physical lows are applied among the agents. Specifically, we build a Voigt system where the agent sensing capabilities are modeled by virtual viscoelastic links (See Figure 1 for an agent-agent interaction model sample). The virtual viscoelastic links are created and detached between the nearest $K$ neighbors, $\mathcal{T}_i(t)$, of each agent, according to particular rules driven by the diversification of the distance metric used in the $k$-NN approach. The flowchart of the control models is illustrated in Fig. 2.
will interact with, we shall first briefly provide insight into

Prior to introducing how each agent selects its

\[ \hat{p}_i = \sum_{j \in T_i(t)} \left( k_{ij}^s (d_{ij} - d_0) \hat{d}_{ij} + k_{ij}^d (v_i - v_j) \right), \quad (2) \]

where \( k_{ij}^s = k_s / \sqrt{d_{ij}} \) and \( k_{ij}^d = k_d \sqrt{d_{ij}} \) are respectively the adaptive spring and damping coefficients with \( k_s \) and \( k_d \) are constants. Whereas, \( d_0 \) represents the equilibrium length of the spring and \( \hat{d}_{ij} \) indicates the direction of the virtual spring force vector.

The computed virtual viscoelastic force, \( \hat{p}_i \), can now be transformed into signals to compute the forward and angular velocities, \( v_i \) and \( \omega_i \) (See (3)). Then, these have to be used to get the velocities of the left and right wheels of the agent, according to (1).

\[ \omega_i = k_\omega \angle \hat{p}_i, v_i = \frac{v_{max}}{\sqrt{|\omega_i| + 1}}. \quad (3) \]

The \( \angle \hat{p}_i \) in (3) is the angle formed by the \( x \) and \( y \) components of the vector \( \hat{p}_i \) and \( k_\omega \) is a gain constant.

The selection of the \( k \)-NN, \( T_i(t) \), among the available neighbors, \( N_i(t) \), is identified according to different methods, which will be discussed in the following subsections.

A. PRELIMINARIES

Prior to introducing how each agent selects its \( k \)-NN that will interact with, we shall first briefly provide insight into the different \( k \)-NN based association distance metrics that are usually used in machine learning and from which we take inspiration to develop our newly proposed aggregation methods

\( k \)-NN is considered as one of the simplest lazy classification/regression algorithm that predicts the output for a new query point according to a similarity measure among its \( k \) closest neighbors in the training dataset [22]. The similarity between points is commonly determined using a distance function; the most used one is Euclidean distance. In literature and based on the properties of data, other sorts of distance measures exist. To this end, we use in our study the cosine distance (CosDist), the angular distance (AngDist), and the Minkowski distance (MinkDist) metrics. The formal definition of these metrics is given below [22]–[24]:

**Definition 1:** For a given feature space, \( S \), of dimensionality \( m \), and having \( A = (x_1, x_2, \ldots, x_m) \) and \( B = (y_1, y_2, \ldots, y_m) \) as two points (feature vectors) in \( S \). Then, the cosine similarity (CosSim) between \( A \) and \( B \) is defined as:

\[ \text{CosSim}_{A,B}^{(m)} = \frac{A(m) \cdot B(m)}{\|A(m)\| \cdot \|B(m)\|}, \quad (4) \]

Their CosDist is:

\[ \text{CosDist}_{A,B}^{(m)} = 1 - \text{CosSim}_{A,B}^{(m)}, \quad (5) \]

And their AngDist is:

\[ \text{AngDist}_{A,B}^{(m)} = \frac{\cos^{-1}(\text{CosSim}_{A,B}^{(m)})}{\pi}, \quad (6) \]

Finally, their MinkDist is:

\[ \text{MinkDist}_{A,B}^{(m)} = \left( \sum_{i=1}^{m} (x_i - y_i)^r \right)^{1/r}, \quad (7) \]

The cosine and angular distance metrics belong to the family of the cosine similarity, which just measures the cosine of the angle between two feature vectors. This family of metrics is mostly used in text mining and information retrieval. The Minkowski distance of order \( r \) is considered as a generalization form of the standard Euclidean metric (case \( r = 2 \)), it is commonly applied in clustering methods.

B. IDENTIFICATION OF THE K-NEAREST NEIGHBORS

In this subsection and basing on the above-highlighted preliminaries, we explain how an agent builds a \( k \)-NN neighborhood topology among all the neighbors that are in its field of view. First of all, each agent should compute its density, \( \rho_i \), on the swarm using a Smoothed particular Hydrodynamic (SPH) density estimation technique as follows [46]:

\[ \rho_i = \sum_{j \in N_i} W(d_{ij}, h), \quad (8) \]

where \( W \) are the kernel functions used to interpolate \( \rho_i \) as the weighted sum of distances over the agent’ neighbors that are within the smoothing length \( h \). In this work, the
kernel functions we used are the following M4 cubic spline functions with \( h = D_r / 2 \) [47]:

\[
W(d_{ij}, h) = \delta \begin{cases} 
\frac{1}{2} (2 - q)^3 - (1 - q)^3, & 0 \leq q < 1 \\
\frac{1}{2} (2 - q)^3, & 1 \leq q < 2 \\
0, & q \geq 2
\end{cases}
\]

(9)

where \( \delta = 10 / (7\pi h^2) \) is a normalized constant and \( q = d_{ij} / h \). Next, each agent should immediately send the calculated density, \( \rho_i \), to its neighbors that are in its communication range. The density received on the other side is denoted by \( \rho_j \).

At this point and upon receiving densities from the neighbors, an agent, \( i \), can build a local neighborhood table as illustrated in Fig.3(a). The table is represented as a matrix of dimension \( |N_i| \times 3 \), where each row, \( B_{ij} = (d_{ij}, \theta_{ij}, \rho_j) \), contains information about the \( j \)th neighbor. Additionally, as structured in Fig.3(b), the agent builds its own 3D vector, \( A_i = (0, \phi_i, \rho_i) \), which contains information about its current distance (\( d_i \)), orientation (\( \phi_i \)) and density (\( \rho_i \)). Here, since there is no distance between an agent and itself, the agent's distance is zeroed.

The agent now uses its built-in local neighborhood table and its own local vector to compute the distance metrics, sorts them from the nearest to the farthest, and then chooses the set of neighbors \( T_i \), that will interact with according to those having the \( k \) nearest distance metrics. These steps are highlighted in Fig.3(c) and Fig.3(d), and are far detailed in the following subsections:

1) A Distance-Weighted based \( k \)-NN (DW-KNN)

This method has been used in our previous \( k \)-NN based aggregation model [21], where each agent applies a weighted distance towards its neighbors. Here, the distance-weighted metric is achieved simply by multiplying the distance \( d_{ij} \) towards neighbor \( j \) by the corresponding received density, \( \rho_j \).

We denote the weighted distance between the agent, \( i \), and its neighbor \( j \) by \( w_{ij} \) where \( w_{ij} = \rho_j \cdot d_{ij} \). Note that this method does not rely on vector \( A_i \) and uses only two features (\( d_{ij} \) and \( \rho_j \)) from vectors \( B_j \).

Now, the robot identifies its distance-weighted \( k \) nearest neighbors (DW-KNN), \( T_i(t) \), that will interact with, by sorting its local neighborhood table in order of the nearest \( k \)-weighted distances, where \( k \in \{1, 2, ..., |N_i(t)| - 1 \} \) refers to how many neighbors should the robot take into account. Note that the cardinality of \( |T_i(t)| \) is \( k \), and \( j \in |T_i(t)| \) : \( w_{ij} \leq w_{i1}, \forall l \in |T_i(t)| \).

2) A Distance-Cosine based \( k \)-NN (DC-KNN)

In this method, the cosine distance metrics are computed between the focal agent and its neighbors. These are computed according to (5) through using the agent own vector, \( A_i \), and each vector, \( B_j \), of the agent local neighborhood table. Her since the vectors have 3 features, then the dimension of the feature space, \( m \), is 3. Therefore, the \( \text{CosDist} \) between \( A_i \) and \( B_j \) is \( \text{CosDist}^{(3)}_{A_i, B_j} = 1 - \text{CosSim}^{(3)}_{A_i, B_j} \).

Similar to the DW-KNN, The set of distance-cosine \( k \) nearest neighbors (DC-KNN), \( T_i(t) \), can now be identified through sorting the agent local neighborhood table in order of the computed nearest \( k \) cosine distances as follows \( j \in |T_i(t)| : \text{CosDist}^{(3)}_{A_i, B_j} \leq \text{CosDist}^{(3)}_{A_i, B_l}, \forall l \in |T_i(t)| \).

3) A Distance-Angular based \( k \)-NN (DA-KNN)

In this method, we use instead the angular distance metric to calculate similarities between the agent and its neighbors. This is also achieved using the agent vector, \( A_i \), and each vector, \( B_j \), of the agent local neighborhood table. According to (6) and having \( m = 3 \), the \( \text{AngDist} \) between \( A_i \) and \( B_j \) is \( \text{AngDist}^{(3)}_{A_i, B_j} = \frac{\cos^{-1} (\text{CosSim}^{(3)}_{A_i, B_j})}{\pi} \).

Then the set of distance-angular \( k \) nearest neighbors (DA-KNN), \( T_i(t) \), is determined via sorting the agent local neighborhood table from the \( k \) nearest angular distance to the farthest ones as follows \( j \in |T_i(t)| : \text{AngDist}^{(3)}_{A_i, B_j} \leq \text{AngDist}^{(3)}_{A_i, B_l}, \forall l \in |T_i(t)| \).

4) A Distance-Minkowski based \( k \)-NN (DM-KNN)

Similarly, we rely on the agent own vector, \( A_i \), and the agent local neighborhood table to identify similarities between the agent and its neighbors. In this method we choose applying the Minkowski distance metric as stated in (7). As consequence, the \( \text{MinkDist} \) between \( A_i \) and \( B_j \) with \( m = 3 \) is

\[
\text{MinkDist}^{(3)}_{A_i, B_j} = \left( \sum_{i=1}^{3} (x_i - y_i)^2 \right)^{1/2}.
\]

Now and alike the above cited methods, to select the set of distance-minkowsky \( k \) nearest neighbors (DM-KNN), \( T_i(t) \), the agent sorts its local neighborhood table by the \( k \) nearest minkowsky distances in such way that \( j \in |T_i(t)| : \text{MinkDist}^{(3)}_{A_i, B_j} \leq \text{MinkDist}^{(3)}_{A_i, B_l}, \forall l \in |T_i(t)| \).

V. SIMULATION RESULTS

A. SIMULATION TOOL, ROBOTIC PLATFORM & EXPERIMENTAL SETUP

The experiments in this work are all done using the ARGoS simulation environment [25]. ARGoS comes with the tools and plugins needed to implement realistic models for 3D physics and many well-known robots such as foot-bots, e-pucks, and kilobots. The ARGoS simulation state is modeled as continuous space and can be updated 10 times per simulated second. A rectangular base of 10 meters long and 6 meters wide is located in the center of the experimental arena. The base is surrounded by four walls and contains no obstacles inside it.

Inside the base, 100 foot-bots robots are initially distributed in random positions and oriented towards arbitrary directions. The simulated foot-bot (See Fig. 4) is a circular, two-wheeled deferentially mobile robot with a diameter of 0.17 m, and approximately of 0.29 m height and 0.147 m wide. In our simulation, the robot can reach a maximum speed of \( v_{max} = 0.10 \text{ m/s} \) and communicate within a communication range, \( D_r = 0.15 \text{ m} \), with nearby robots using the range and bearing (RAB) communication device.
The special feature of the RAB device is that, upon receiving a message from a sender, the robot is able to locally measure both the range and bearing measurements. To provide a realistic behavior to the RAB device, the measurements can be noised using noise models. In our simulation, a Gaussian noise model of form \( \mathbf{N}(0, 0.01) \) is added to the range and heading measures. Moreover, we consider that a number of packets can be lost during a communication phase by setting the probability packet loss to 3%.

Table 1 summarizes the different constants and parameters related to the virtual viscoelastic model in (2), the motion computation model in (1) and (3), and the Minkowski distance order used in (7).

### B. ARGOS RESULTS & DISCUSSION

With all the settings presented in the previous sub-section, we conducted, for each aggregation method and while starting from the same initial distribution of 100 foot-bots robots, an experimental simulation for a total duration of 3000 time steps each.

Figure 5 illustrates at different time steps a comparative representation of one particular simulation of each distance \( k \)-NN aggregation method. The Left column of the figure, from (A) to (D), shows specifically snapshots obtained from the ARGoS simulator while performing the DW-KNN, the DA-KNN, the DC-KNN, and the DM-KNN, at time steps: \( t = \{0, 500, 1500, 3000\} \) and with taking into account different values of \( K \) (\( K=2 \) (A), \( K=3 \) (B), \( K=4 \) (C), and \( K=5 \) (D)) in the \( k \)-NN method. The right column of the figure, from (E) to (H), shows the corresponding density spatial distributions of the robots obtained from applying a Gaussian-based kernel density estimation technique among the robots’ positions. The grids generated by these density spatial distributions correspond only to the minimum bounding areas of the simulated environment that enclose all the robots swarm. The grids are marked in different colors dependent on the robots’ actual locations and their \( k \)-nearest neighbors. The robots’ positions in the grid are represented as dark circles. Colors in dark blue represent positions or zones where no robots are located in the corresponding \( x \) and \( y \) coordinates and therefore no neighbors are available in those zones. Red

---

**Figure 3.** Identification of the distance based \( k \)-NN set \( (T_i) \). (a) The agent local neighborhood table at time \( t \) as a set of vectors \( B_{i,j} \in \mathcal{N}_i = (d_{ij}, \theta_{ij}, \rho_{ij}) \).

(b) The agent own vector \( A_i = (0, \phi_i, \rho_i) \) at time \( t \). (c) Computing the distance metrics between \( A_i \) and \( B_{i,j} \) according to the aggregation method using the appropriate equations: Green colour indicates our previous method (DW-KNN), where there is no need for vector \( A_i \) to calculate the metric; Blue sky colour indicates the new proposed methods relying on both vectors \( A_i \) and \( B_{i,j} \). (d) Sorting the computed distance metrics by the \( k \) nearest distances. Then the \( k \) related neighbors among all the agent’ neighbors \( (\mathcal{N}_i) \) constitute the set \( (T_i) \).

**Figure 4.** The simulated foot-bot robot.

**Table 1. Parameters and constants used in the experimental simulations**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>Inter-wheels distance</td>
<td>0.14 m</td>
</tr>
<tr>
<td>( \omega_{max} )</td>
<td>Maximum angular speed</td>
<td>180 °/s</td>
</tr>
<tr>
<td>( k_\omega )</td>
<td>Angular speed gain</td>
<td>1.5 °/s</td>
</tr>
<tr>
<td>( d_0 )</td>
<td>Equilibrium length of the spring</td>
<td>0.68 m</td>
</tr>
<tr>
<td>( k_d )</td>
<td>Damper gain constant</td>
<td>1.25 force unit</td>
</tr>
<tr>
<td>( k_s )</td>
<td>Spring gain constant</td>
<td>1.9 force unit</td>
</tr>
<tr>
<td>( r )</td>
<td>Minkowski distance order</td>
<td>1.5</td>
</tr>
</tbody>
</table>

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see [https://creativecommons.org/licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/).
FIGURE 5. ARGoS Snapshots (Left from (A) to (D)) for the different k-NN aggregation methods with the corresponding robots spatial density distributions (Right from (E) to (H)) during different simulation time steps.
colors represent zones with the higher numbers of \( k \)-nearest neighbors surrounding the robots.

As it can be seen, at different stages of the ARGoS simulation and by varying only the value \( k \) of the \( k \)-NN approaches, different aggregations seem to emerge from all the proposed \( k \)-NN aggregation methods. The results of the final-stage simulation show that, in almost all the cases values of \( k \), the new distance-based \( k \)-NN aggregation methods used in this study, namely the DA-KNN, the DC-KNN, and the DM-KNN achieve better aggregations than our previous DW-KNN method. This can be much noticeable in the corresponding Gaussian kernel density estimation grids shown on the right of Fig. 5, where the higher the zones are dense (red color areas), the best the aggregation is. It is also seen that for the given \( k \)-NN aggregation method, the greater the value of \( k \), the best we achieve better aggregations.

Furthermore, to compare the performance of the studied approaches for the above simulations, we calculated for each particular simulation the surface or the Area of the Minimum Bounding Rectangle (AMBR) that encloses the entire robots. The corresponding results are plotted in Fig.6 as radar charts, in which the axes represent seven different stages of the simulation \( t = \{0, 500, 1000, 1500, 2000, 2500, 3000\} \).

![Radar charts representing the area of the minimum bounding rectangle (AMBR) enclosing the entire swarm at different seven time steps of the simulation](image)

**FIGURE 6.** Radar charts representing the area of the minimum bounding rectangle (AMBR) enclosing the entire swarm at different seven time steps of the simulation.

It can be seen that for almost every \( k \) of the studied method, the colored zone covered by the AMBR of the newly proposed distance aggregation methods (the DA-KNN, the DC-KNN, and the DM-KNN) are practically less than the one obtained from that of the DW-KNN. However, those colored zones are keeping decreasing by increasing the value of \( k \). Notice also that the AMBR of all the methods converges nearly to the same values in case of \( k = 5 \). This is as expected, forming a mesh of more than two robots \( (k > 2) \), will attract more robots to their group center and therefore the resulting aggregation zone will be denser.

**VI. PERFORMANCE ANALYSIS**

This section aims to assess more accurately the performance of the proposed aggregation methods in an intuitive manner. Since the objective is to aggregate \( N \) initially dispersed robots into a compact zone in a somewhere area of the environment, we are therefore looking to minimize the dispersion quality of the swarm. While different dispersion metrics can be used in the literature, we adopt a modified version of the second moment of disks [26] as a metric to measure the dispersion quality of our swarm. In the following subsections: we present the metric, we analyzed the performance of our methods in normal circumstances, and we studied the effect of sensory noise and the effect of a number of faulty robots on the performances of the methods.

**A. DISPERSION METRIC**

Following Gaussi et al. [26], we adopt a modified version of the second moment of the robots to measure the robots dispersion quality [21]. First, we measure the dispersion of each robot, \( i \), as follows:

\[
\text{Disp}_i(t) = \frac{1}{4\pi r_s^2} \sum_{i=1}^{K} \left\| \bar{p}_i(t) - \bar{p}_i(t) \right\|^2, (10)
\]

where \( r_s \) is the robot’s radius, \( \bar{p}_i(t) \) is the robot’s position at time \( t \), and \( \bar{p}_i(t) = \frac{1}{N} \sum_{j=1}^{N} p_j(t) \) is the position of the center of mass of the robot’s group at time \( t \).

The dispersion measure of the entire swarm at time \( t \), which we denote \( F_{\text{disp}} \), can be then averaged over the number of the robots as follows:

\[
F_{\text{disp}}(t) = \frac{1}{N} \sum_{i=1}^{N} \text{Disp}_i(t). (11)
\]

**B. ANALYSIS OF THE METHODS IN NORMAL CIRCUMSTANCES**

Following the setup described in Section V-A, we conducted, for each distance based \( k \)-NN aggregation method, 25 runs of a duration of 3000 time step each (300 seconds). For every 10 time steps of a given run, we collected data from the ARGoS simulator.

Figure 7 shows, for each value of \( k \), results of the dispersion metric that are obtained from performing the proposed distances based \( k \)-NN methods on 100 robots. The scattered symbols in the figure represent the mean values of the metric obtained from 25 runs with different initial configurations of the robots. While the colored dashed lines represent a fitting curve of the form \( Ae^{-Bt} + C \).

As it can be seen from the plots, all the methods degrade exponentially to a lower \( F_{\text{disp}} \) value. Note that as stated in [26], a lower value of \( F_{\text{disp}} \) indicates a better sign of aggregation. Therefore, as the newly proposed distances based \( k \)-NN aggregation methods have lower \( F_{\text{disp}} \) values in comparison to the DW-KNN method, so their aggregation performance is better than the previous DW-KNN method. While this is particularly noticed in cases where \( k = \{2, 3, 4\} \), it is
observed that when $k = 5$, all the methods converge slightly to the same dispersion metric value. Also, as $k$ increases, the aggregation quality of all the methods start to improve. However, in all cases of $k$, the DM-KNN method has a faster decay in comparison to the other methods, meaning that it has the best performance in terms of convergence among the other aggregation methods.

On the other hand, the quality performance of the DA-KNN and the DC-KNN methods is almost the same for all the case studies. This is as expected due to the fact that both approaches rely on the $\cosSim$ measure, which in our case, sorting the neighbors according to the $k$-nearest $\cosDist$ and $\cosAng$ will always give the same order and therefore the same set of interacting neighbors, $T_i(t)$. 

C. EFFECT OF SENSORY NOISE

In this subsection, we assess the performance of the proposed $k$-NN based aggregation approaches while the readings of the robots RAB sensors are noised. As in the previous analysis, we remain the settings described in Section V-A the same. However, the range and bearing readings of the RAB sensors are noised using a Gaussian distribution of the form $\mathcal{N}(0, \sigma)$, with this time taking into consideration different values for the standard deviation $\sigma$. The values taken into account in our study are $\sigma = \{0.2, 0.4, 0.6, 0.8\}$. For each value of $\sigma$, 25 trials per simulation experiment are performed for every $k$-NN aggregation method.

Figure 8 plots, for each value of $k$, the aggregation quality, $F_{\text{disp}}(t)$, of each method, in the form of error bars with respect to the different $\sigma$ values. Each error bar represents the $\pm$ standard error of $F_{\text{disp}}(t)$, which are obtained from the 25 trials with different initial configurations of the 100 robots. The dashed lines show a linear fit to the error bars.

An analysis of this figure indicates that all the aggregation methods are, as expected, affected by the increase of noise in the readings of the RAB sensors. We can see that for every value of $k$, an increase in $\sigma$ causes an increase in $F_{\text{disp}}(t)$ for all the studied $k$-NN aggregation approaches. Furthermore, in almost all cases of $k$, the DM-KNN aggregation method works better than the other methods even when $\sigma$ increases. However, we can notice that if $k$ increases, the quality of the aggregation of the methods converges almost to the same $F_{\text{disp}}(t)$ for all $\sigma$ values. This can be very noticed in the case where $k = 5$. Alike in the normal circumstances case studies, the DA-KNN and the DC-KNN methods still perform equally in all the case studies of $\sigma$ and $k$. This is as expected because also both approaches rely on the $\cosSim$ measure. Therefore, sorting the neighbors according to the nearest $k$ distances ($\cosDist$ and $\cosAng$) always gives the same order, which results in having the same set of interacting neighbors ($T_i(t)$). In addition, as $\sigma$ increases, the DW-KNN aggregation method starts to produce better results than the DC-KNN and DA-KNN approaches in almost all $k$ cases. This is because the DW-KNN approach uses only one noisy feature, namely $d_{ij}$, out of two compared to the other two approaches, where two noisy features ($d_{ij}$ and $\theta_{ij}$) are involved. As a result, the increase in noise in two features will completely affect the performance of the DA-KNN and the DA-KNN compared to the performance of the DW-KNN.

D. EFFECT OF FAULTY ROBOTS

In this sub-section, we investigate how the aggregation methods are tolerant to faulty robots. This is achieved by measuring the aggregation quality for a swarm of 100 robots within a number of partially faulty robots. We mean by a partially faulty robot, that one of the robot’s component is not stopping completely to perform, rather one of its sub-components becomes a failure. This kind of faults has been reported to

![FIGURE 7. Dispersion metric results obtained from the different proposed distances based k-NN methods performed on 100 robots. In all plots, the x-axis represents the evolution of time step $t$. Each scattered symbol represents the median values obtained from 25 runs with different initial configurations of robots. The dashed lines are the corresponding fitting curves of the form: $Ae^{-Bt} + C$.](https://example.com/figure7)

![FIGURE 8. Effect of noise, $\sigma$, in the RAB sensors readings on the performance of the swarm, $F_{\text{disp}}(t)$, performing the proposed $k$-NN aggregation methods. Noise has a Gaussian distribution of the form $\mathcal{N}(0, \sigma)$. 25 runs per each setting with 100 robots were performed and averaged. Each error bar represents the $\pm$ standard error and the dashed lines are linear fits to the error bars.](https://example.com/figure8)
be having more effect in damaging the overall collective behaviors of robotics swarms systems while compared to a complete fault type [48]. Here, we particularly evaluate the effect of a partially motor failure type on the aggregation performance of the methods. In this type of faults, one motor - the left or the right - fails to perform during certain periods of the total time running. To this end and by following the same setup of Section V-A, the aggregation quality of the methods is assessed within 10%, 20%, 30%, and 40% of robots that we assume they are been affected by a partial motor failure. Here, the injection of this fault type on the robots’ motor encoders is programmed during the ARGoS simulation in two separate intervals (\( t \in [500, 1200] \) and \( t \in [1500, 2200] \)) of the total time simulation (\( t = 3000 \)). Whereas, the identification of the robots chosen to be faulty is done in a random manner. As a consequence, the randomly programmed faulty robots will perform about 47% of their total running time simulation with a partial motor fault type.

![Figure 9](https://example.com/figure9.png)

**FIGURE 9.** Effect of faulty robots (%) on the swarm aggregation quality. 25 runs for each aggregation method per each setting are averaged.

Figure 9 illustrates the results obtained from performing 25 simulations runs per percentage of faulty robots for each aggregation method and each value of \( k \). The results correspond to the performances of the methods of all the swarm including both normal and abnormal robots over the overall time simulation that involves the two faulty studied intervals. As shown in the figure, the performance quality of each distance-based aggregation \( k \)-NN method does not seem to be affected by the number of faulty robots. It is seen that for each \( k \) case study, the aggregation quality per distance \( k \)-NN method remains almost the same even when the number of partially faulty robots increases. More precisely, and as in the previous analysis sub-sections, the performance quality of the DM-KNN aggregation method in all \( k \) cases studies is superior to the other methods performance qualities whatever the number of faulty robots is. Also, as \( k \) increases, the performance quality, \( F_{\text{disp}}(t) \), of each aggregation method improves regardless of the number of faulty robots. However, the aggregation qualities of all the methods start to converge slightly as \( k \) increases. In addition, both the DA-KNN and the DC-KNN still perform at the same level by \( k \) case study regardless of the number of faulty robots. Moreover, their aggregation quality remains better than the WD-KNN method in all \( k \) case studies.

**VII. CONCLUSION AND FUTURE WORKS**

Self-organized aggregation within robot swarms systems has been a matter of interest in the last decade. Recent studies in addressing such behavior in such type of multi-robotics systems have been inspired from the topological distance metric known in the collective behavior of biological social animals (e.g. birds flocks and fish schools). However, using the density of the swarm as an additional property to enhance the swarm aggregation performance has been only incorporated recently [20], [21] through an aggregation method called a DW-KNN.

On the other hand, in machine learning, distance metrics such as Euclidean, Cosine, Angular, Minkowski, and others can be used based on the property of data to improve \( k \)-NN classification/regression algorithms. So far, such metrics have not yet been applied to calculate distances between neighboring robots at real-time. Therefore, in this paper, three topological aggregation methods based on the Cosine, Angular, and Minkowski distance functions have been proposed to specifically improve the performance quality of the DW-KNN aggregation method. The three methods, which are named: the DA-KNN, the DC-KNN, and the DM-KNN have been implemented in a swarm of 100 foot-bots robots using the ARGoS simulator. The particularity of these methods is that they are able to achieve aggregation using the robots’ range and bearing sensors readings to measure distance metrics among robots neighbors. More precisely, with these methods, a robot can identify its neighborhood set that will interact with to form a virtual viscoelastic mesh that leads to achieving aggregation.

A comparative study had been first done to assess the performances of the three methods with that of the DW-KNN approach using the area of the minimum bounding rectangle (AMBR) that enclose the entire swarm as a metric. Results within the AMBR showed that the newly proposed aggregation methods had better performances than the DW-KNN method. However, the overall AMBRs of all the methods started reducing gradually when \( k \) increases.

Furthermore, to more accurately evaluate the performances of all the methods, analysis studies had been done to minimize the dispersion quality of the swarm using a dispersion metric. Analytical studies within this metric involved accessing the performances of the methods under normal circumstances, in the presence of sensory noise and finally in the presence of partially faulty robots. Results in normal circumstances within the dispersion metric illustrated a better performance of the newly proposed distances based \( k \)-NN aggregation methods in comparison to the DW-KNN. On the other hand, the overall performance results of the on-studying aggregation methods in the presence of noise in the robots’ range and bearing sensors showed a gradual sensitivity of
the methods to these noisy sensors. However, the methods are still robust to achieve aggregation in such circumstances. The DM-KNN remained also the better method under such circumstances in comparison to the other methods. However, while progressively increasing the noise in the robots’ sensors, the DW-KNN aggregation method starts producing better results than the DC-KNN and DA-KNN approaches. Also, under abnormal robots where the motor encoders capabilities of a number of robots become partially defective, it is shown that all aggregation methods remain robust against this type of faults. Furthermore, results in all the studies carried out in this work, report that the aggregation performance of the DM-KNN method outperforms all the other methods.

As future work, we are seeking to see how well the new \( k \)-NN methods will perform against existing methods of aggregations, other than the \( k \)-NN-based one. Also, we are looking forward to investigating the effect of more distance-based machine learning metrics, particularly on the aggregation performance of robotics swarm systems and generally in addressing collective behaviors within such type of systems as well as multi-agent systems. We believe that the present work could be a first step towards applying such metrics in improving various collective behaviors studies in the literature. We are planning also to investigate the possibility of the implementation of such distance metrics in an on-board hardware sensor to accelerate data processing.

REFERENCES


***