Swarm Flocking using Optimization for a Self-Organised Collective Motion

Mazen Bahaidarah[a,b], Fatemeh Rekabi-Bana[c], Ognjen Marjanovic[a], Farshad Arvin[c,*]

[a] *Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, M13 9PL, UK*
[b] *Department of Electrical and Electronic Engineering, King Abdulaizi University, Jeddah, Saudi Arabia*
[c] *Swarm & Computational Intelligence Laboratory (SwaCIL), Department of Computer Science, Durham University, UK*

Abstract

Collective motion, often referred to as flocking, is a prevalent behaviour observed in nature wherein large groups of organisms move cohesively, guided by simple local interactions, as exemplified by bird flocks and fish schools. Collective motion behaviour can be achieved by applying virtual physical interaction inspired by elastic materials to determine the required attraction and repulsion forces between the agents in a swarm robotic system. However, it is necessary to apply virtual interaction efficiently to prevent undesirable swarm fluctuation, slow alignment, and excessive energy consumption. This paper presents a novel Optimised Collective Motion (OCM) algorithm, which exerts viscoelastic interaction between robots. Moreover, the algorithm's parameters are tuned automatically by Particle Swarm Optimisation (PSO) to achieve: (i) minimum control effort, (ii) fast alignment, and (iii) robustness against noise. Simulation results establish the proposed algorithm outperforms the previous methods, achieving better alignment and being more robust in the presence of measurement noise. Furthermore, the effectiveness of the OCM algorithm is validated by real-robot experiments performed on a swarm of six miniature mobile robots. These experiments demonstrate the practical capabilities of the proposed algorithm.

Keywords: Swarm Robotics, Collective Motion, Bio-inspired Swarm, Optimisation, Virtual Viscoelastic links

1. Introduction

Collective Motion (CM), is a noticeable phenomenon in nature which is observable in various areas from biological systems such as birds flocking, insect swarms, and schools of fish to molecule movement [1]. Such organisms are able to coordinate their motion collectively and improve their performance. For instance, a school of fish can collectively evade predators while maintaining the structural integrity of the group [2]. Additionally, flocks of birds, especially in energy-efficient formations such as the V-Shape, can traverse vast areas with considerably less energy consumption per individual [3]. Various mathematical algorithms are developed inspired by that appealing natural behaviour to solve real-world challenges in swarm robotics. Swarm robotics is the study of how large groups of simple, small, and commonly identical robots can collectively accomplish complex tasks that would be impossible for an individual robot to accomplish alone. Swarm robotics offers substantial advantages in terms of robustness, scalability, and flexibility [4] when real-world applications are considered.

Collective motion behaviour in the swarm robotics field has been deployed across a broad array of challenges, showing remarkable results in diverse areas such as precision agriculture [5], search and rescue operations in disaster zones [6], and exploration of unknown environments [7]. A self-organised collective motion can be achieved when a swarm of robots moves cohesively, leveraging local interactions to exchange crucial information such as positions, velocities or angles.

The cornerstone for flocking behaviour modelling in artificial systems was established by Reynolds [8]. In that model, cohesion is achieved by attraction between the robots and repulsive interaction is considered to prevent the agents from collision. Also, robots' alignment ensures the entire swarm moves in a particular direction, maintaining the same speed. This general model assumes that the focal robot must be aware of its neighbours' headings to facilitate flocking in a noise-free environment. The Self-Propelled Particles (SPP) approach is another method aiming for robots flocking inspired by statistical physics properties.

The SPP model exploits the virtual attractive and repulsive forces between swarm entities to create a self-organised collective motion swarm [9]. Notably, two distinct approaches for exhibiting flocking behaviour exist, which are the alignment rule and the position-based rule. The alignment rule involves explicit sharing of angle information amongst agents to reach a consensus on their headings [10]. However, robots which use that approach need more sophisticated measurement and communication instruments to share their orientations reliably.

Therefore, the position-based rule, which allows the focal agent to detect merely the positions of its neighbours, emerges as a more cost-effective choice. Swarm cohesiveness and alignment are achieved utilising only relative position data [11]. Al-

[*] Corresponding author. Fatemeh Rekabi-Bana fatemeh.rekabi-bana@durham.ac.uk

Email addresses: mazen.bahaidarah@manchester.ac.uk (Mazen Bahaidarah), fatemeh.rekabi-bana@durham.ac.uk (Fatemeh Rekabi-Bana), ognjen.marjanovic@manchester.ac.uk (Ognjen Marjanovic), farshad.arvin@durham.ac.uk (Farshad Arvin)

though such collective algorithms are cost-efficient, the swarm cannot reach reasonable robustness and respond very slowly if it depends on relative position data merely. This paper introduces a novel collective motion control method developed based on viscoelastic interaction between the agents to mitigate the fluctuations much faster and improve the swarm performance considerably. This is accomplished by using the relative position and velocity among the neighbours to apply the interactive force based on both spring and damping effect and reduce the energy level and reach the stable point as fast as possible. Furthermore, the Particle Swarm Optimisation (PSO) is applied to obtain the best control parameters according to the objective function which includes the interactive forces, alignment, and response time to obtain a reasonable response regarding these three vital criteria. The feasibility of employing PSO to optimise the performance of collective motion has been demonstrated previously [12]. That idea is extended to optimise more parameters and consider measurement noise according to the proposed control architecture. As a result, the main contributions of this paper are summarised as follows:

- Introducing a novel swarm control framework considering viscoelastic interaction to enhance the swarm's stability and cohesiveness and diminish the fluctuations much faster.

- Formulating a multi-objective optimisation problem that ensures convergence and automates the tunning procedure for the control parameters.

- Proposing a Monte Carlo simulation scheme for robust performance evaluation in the presence of measurement noise.

The rest of the paper is structured as follows: Section 2 presents a literature review pertinent to the work under consideration. Section 3 elaborates on the theoretical underpinnings of the collective motion algorithm. Section 4 illustrates the theoretical aspects of Particle Swarm Optimisation (PSO) and details the design of the cost function. In Section 5, the experiments and implementation in both simulation and real-world contexts are presented. Section 6 comprises an analysis of the results obtained from both simulations and real-world implementations. The implications of altering significant parameters in the OCM model are discussed in Section 7. Finally, Section 8 concludes the research and outlines possible directions for future work.

2. Related Work

The Self-Propelled Particles (SPP) concept was exploited in the Standard Vicsek Model (SVM) [10] which is one of the earliest methods in flocking. Although that research was aimed at investigating the effect of noise and particle size on transitioning between the ordered and disordered state, it shows promising results employing a velocity alignment rule whereby each particle adjusts its trajectory towards the neighbours' headings average and shares its orientation with them. In spite of the fact that applying velocity alignment demonstrates reasonable results, Couzin et al. presented another collective behaviour model in three-dimensional environments. In that study, three different interactions were considered to emulate the animal groups' transitions between various structures. Those three interactions are applied in distinct zones to determine the attraction, alignment, and repulsion between the members.

The above foundational flocking models inspired many studies in swarm robotics [13, 14, 15]. For instance, an establishment of how informed agents can direct a uniformed swarm in a precise direction using the velocity alignment rule is proposed in [16]. The agents' velocities are updated according to a Laplacian-based model which regulates each agent's velocity difference with its neighbours. That study inspired real-world experiments in [17]. However, such frameworks depend on exchanging both velocity and alignment data that demands considerable onboard processing which is a challenge for small robots with limited power and processing sources.

Accordingly, several studies have been dedicated to implementing collective motion behaviour without relying on orientation information. For instance, velocity alignment is achieved implicitly by applying pairwise repelling forces to move the particles in [18]. Furthermore, exploiting other approaches such as position-based attraction and repulsive force determination [19], the inelastic collision between isotropic agents [20], and other collective control strategies [21] investigate different approaches which are independent of orientation sharing between the agents and discuss the advantages of such methods.

Therefore, it is beneficial to reduce the exchanged information between robots to minimise the hardware complexity and cost of the robots [22]. Moreover, according to the aforementioned studies, it is evident that alignment without explicit orientation exchange can reduce the entire swarm's energy consumption. Another similar approach was developed by Ferrante et al. [11, 23] as a state-of-the-art flocking model known as Active Elastic Sheet (AES). That method suggests elastic interaction based on relative position to achieve collective motion. The AES performance is established in several studies from different perspectives. For instance, the effect of network architecture [24], the robustness against measurement noise [25], and the behaviour in the presence of external force to guide the swarm in a particular direction [26] are investigated to demonstrate the AES capability in various scenarios.

Moreover, other papers try to reveal other aspects of the AES, such as its performance considering obstacle avoidance feature [27, 28] and scalability while real robots are used for the experiments [29], [30]. However, the research didn't consider other performance criteria such as required control effort, response time, and closed-loop response considering measurement noise. Although the AES-based algorithms present promising results, using virtual elastic interaction merely produces fluctuations that can cause the entire swarm to become unstable and insufficiently robust to maintain the system's output in a reasonable bound when the measurements include noise. However, viscoelastic interaction can mitigate such fluctuations and improve stability and robustness simultaneously. Therefore, it is anticipated that exploiting the properties of viscoelastic links could significantly enhance flock motion while concurrently preserving the stability of its formation.

In addition to the model's structure describing the collective behaviour, the parameters have a vital contribution to the final results. Those parameters are generally tuned empirically, although the optimisation techniques can enhance the performance significantly. Particle Swarm Optimization (PSO) is a prevalence evolutionary algorithm that is proposed by Kennedy and Eberhart [31] and inspired by the flocking behaviour of birds when they seek a source of food. The PSO algorithm is popular in the research domain because of its simplicity and low computational processing requirement. In the robotic field, it has been extensively used for optimising a variety of problems. For instance, in [32], PSO is utilised to improve a three-dimensional path planning algorithm for UAV formation. In another study [33], the motion stability and efficiency of a fruit-picking manipulator are optimised by utilising an improved multi-objective PSO. Despite the simplicity and robustness of PSO, a significant challenge lies in the potential for a local minima trap. That issue can be circumvented and the exploration and exploitation capabilities will be improved by employing several techniques to evade the local minima trap which are considered in this paper.

The collective behaviour of swarm systems has been optimised via the PSO algorithm in a number of studies, specifically through the tuning of motion controller parameters. For instance, In [34], the PSO algorithm is used to select the proper control parameters of a quadrotor flock that collectively moves to seek a targeted zone with and without obstacles. In another research [35], the PSO algorithm was applied to allocate the leader task to proper UAVs divided into two clusters. Therefore, it is evident that the PSO algorithm has a substantial capability to significantly enhance the collective performance of swarm systems. In [26], an optimisation algorithm called Tabu Continuous Ant Colony System (TCACS) [36] is used to tune the parameters of the AES model by minimising the force amongst the robots and maximising the alignment of the robots' headings. The results showed better performance in comparison with the original parameters in [11]. In another study [12], the collective motion of the AES model is improved by employing the PSO algorithm. The cost function is a minimisation problem, which was focused on: i) the virtual forces between the swarm individuals, ii) the alignment error of the whole swarm, and iii) the convergence time. The results exhibited a significant performance in terms of collective motion behaviour and stability of the swarm shape that outperformed the original studiy [11] and the study optimised by TCACS [26]. However, those works did not consider the measurement noise and its impact on the collective motion behaviour.

3. Collective Motion

This section describes the theoretical background of the collective motion algorithm that is proposed in this paper.

3.1. Active Elastic Sheet (AES)

The state-of-the-art AES model utilised the elasticity-based mechanism, which introduced the elastic interactions between the swarm members. AES is a position-based approach in which each robot within the swarm is equipped with a virtual spring-like connection to its neighbours, thereby sharing only relative positional information [23]. The overall elastic force $\vec{F}_i$ exerted by i^{th} robot is originated by its interaction with its neighbours S_i. The rotational speed of the robot is determined by the angle between this force vector and the robot's heading direction. If the force vector is not parallel to the robot's heading, a rotational motion is induced, while a parallel orientation between the force and heading results in linear motion.

Figure 1 shows two of the agents in the swarm and the resultant force $\vec{F}_i$ acting on one of the agents determined from the interaction with all neighbours. According to Figure 1, it is evident that the resultant force has a component which is perpendicular to the robot's current direction, causing rotation and eventually resulting in an alignment between the agents.

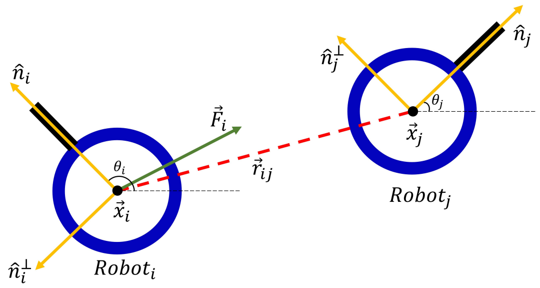

Figure 1: The schematic description of elastic interaction .

3.2. Optimised Collective Motion (OCM)

As mentioned in the previous section, virtual forces induced among the robots serve as a primary determinant for collective movement and swarm shape preservation. Considering a flexible swarm structure and spring connections to model the robots' interactive behaviour causes undesirable fluctuations. Such unwanted fluctuations can make the system unstable.

To mitigate this issue, a virtual viscoelastic mechanism is deployed between the swarm individuals. A viscoelastic link between two robots comprises a spring with a specific stiffness k and a damper with a coefficient c, arranged in a parallel configuration, as depicted in Figure 2. It is important to note that the damper coefficient and spring stiffness values remain constant across all the established robot links. Thus, the regulation of the compression and stretching of the viscoelastic links enhances the motion smoothness and stability of the entire swarm. Furthermore, the Optimised Collective Motion (OCM) model benefits from the energy dissipation property inherent in the damper to diminish the fluctuations and reach a stable condition much faster.

At the beginning of an experiment, a group of N robots move as a swarm in a two-dimensional arena. The swarm's individuals are located in predefined positions as a square shape

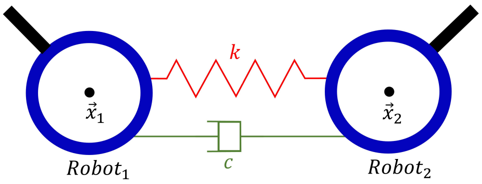

Figure 2: A single spring-damping system.

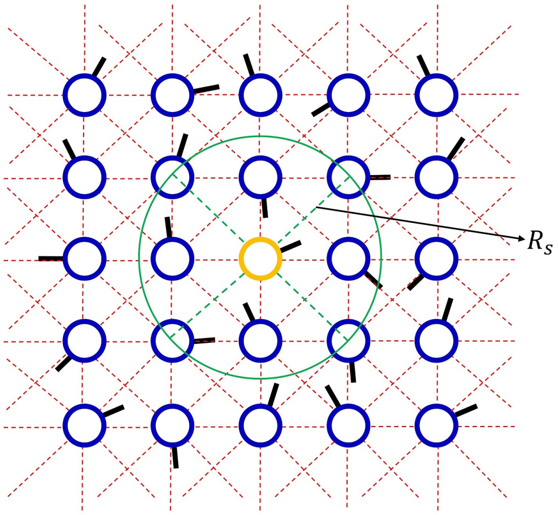

Figure 3: The Nearest-Neighbour network for a part of a swarm of 100 robots. Each blue circle denotes a robot body with a small black line representing the heading. Each red dotted line denotes the natural length between the robots. The green circle shows the sensing range of the focal robot (orange) by considering the sensing radius R_s.

at $t = 0$ s. The virtual force $\vec{F}_i$ applied to the i^{th} robot constitutes the cumulative impact of all viscoelastic links associated with its neighbours S_i. This study applies a Nearest-neighbour (NN) network for interaction topology, wherein each robot connects to its nearest neighbours within a specified sensing radius R_s, as illustrated in Figure 3. Robots in the middle of the NN network have the highest number of connections (eight links), whereas robots in the corner have three links and robots on the side have five links. Therefore, the virtual force $\vec{F}_i$ of the i^{th} robot can be mathematically formulated as follows:

$$\vec{F}_i = \sum_{j\in S_i} -\frac{k}{l_{ij}}(\|\vec{r}_{ij}\| - l_{ij})\frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\|} - (c\,\vec{v}_{ij})\,, \tag{1}$$

$$\vec{r}_{ij} = \vec{x}_j - \vec{x}_i\,, \tag{2}$$

$$\vec{v}_{ij} = \vec{v}_j - \vec{v}_i\,. \tag{3}$$

where, l_{ij} is the natural length that connects robots i and j, and $\frac{k}{l_{ij}}$ is the spring constant. S_i denotes the neighbours of the i^{th} robot and $\vec{r}_{ij}$ represents the distance between i^{th} and j^{th} robots. c is the damper coefficient, where $c > 0$. The velocity vector $\vec{v}_i$ of the i^{th} robot can be obtained by multiplying the constant speed v_0 by the unit vector $\hat{n}_i$. The variable $\vec{v}_{ij}$ represents the difference in the velocity between the current robot i with respect to its neighbour j. The position $\vec{x}_i$ and heading θ_i of the i^{th} robot can be calculated as follows:

$$\dot{\vec{x}}_i = v_0\,\hat{n}_i + \alpha[(\vec{F}_i + D_r\hat{\xi}_r)\,.\,\hat{n}_i]\hat{n}_i\,, \tag{4}$$

$$\dot{\theta}_i = \beta[(\vec{F}_i + D_r\hat{\xi}_r)\,.\,\hat{n}_i^{\perp}] + D_\theta\xi_\theta\,, \tag{5}$$

$$\hat{n}_i = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}, \tag{6}$$

$$\hat{n}_i^{\perp} = \begin{bmatrix} \cos(\theta_i + \frac{\pi}{2}) \\ \sin(\theta_i + \frac{\pi}{2}) \end{bmatrix}. \tag{7}$$

According to Eq. (4) and Eq. (5), a constant speed v_0 is imposed on each robot. The parameters α and β are inverse transitional and rotational that control the linear and angular speed of the i^{th} robot, respectively. $\hat{n}_i$ which is determined by Eq. (6), is a unit vector pointing to the same heading direction of the robot i, and $\hat{n}_i^{\perp}$ is a unit vector pointing perpendicular to it represented in Eq. (7). In this approach, two types of random processes are considered. The first, actuation disturbances, denoted as $D_\theta\xi_\theta$, embodies the fluctuations of the robot motion by adding it to the robot's heading $\dot{\theta}_i$. The second type, measurement noise, expressed as $D_r\hat{\xi}_r$, represents the inaccuracies in force determination because of position measurement noise. ξ_θ is a random variable with normal probability distribution, whereas $\hat{\xi}_r$ is a randomly oriented unit vector. D_θ and D_r are coefficients used as a scale to control the noise and disturbance level.

Angular alignment among swarm members plays a pivotal role in maintaining the cohesiveness of collective motion. A key metric parameter to represent the alignment status of the robots in this study is the degree of alignment, denoted as ψ. In scenarios where all robots are aligned in a unified direction, $\psi \approx 1$. Conversely, in a state of disorder, wherein the robots exhibit a random orientation, $\psi \approx 0$. The mathematical formulation to compute the degree of alignment is as follows:

$$\psi = \frac{1}{N}\left\|\sum_{i=1}^{N}\hat{n}_i\right\|, \tag{8}$$

where, N is the swarm size and $\hat{n}_i$ represents the unit vector of the i^{th} robot. ψ can be obtained by taking the norm of the mean value of all the robots' heading values. The Pseudocode that represents the mechanism of the proposed OCM model is shown in Algorithm 1.

4. OCM Parameters Optimisation

This section demonstrates the optimisation process based on the PSO algorithm.

4.1. Particle Swarm Optimisation (PSO)

PSO is an evolutionary optimisation technique which is applicable to a wide range of problems. The heuristic search makes such optimisation algorithms more robust regardless of the objective function structure. Applying the PSO algorithm

Algorithm 1 OCM Model

```
Obtain the optimal values of $(\alpha,\beta,k,c)$ using PSO algorithm.
Set swarm size $N$, arena length $L$, sensing range $R_s$, contant speed $v_0$, and simulation time $T_{sim}$.
Initialise robots' positions $x$ in a square shape and random headings $\theta$.
Obtain velocity vector $\vec{v}$ for each robot.
for $step = 1, ..., T_{sim}$ do
  for $i = 1, ..., N$ do
    Find neighbour list $S_i$ of the $i^{th}$ robot.
    Calculate $\hat{n}_i$ and $\hat{n}_i^{\perp}$ using Eq. (6) and Eq. (7).
    for $j = 1, ..., S_i$ do
      Calculate $\vec{r}_{ij}$ using Eq.( 2).
      Calculate $\vec{v}_{ij}$ using Eq.( 3).
      Calculate $\vec{F}_i$ using Eq.( 1).
    end for
    Update new position $x_i$ using Eq.( 4).
    Update new heading $\theta_i$ using Eq.( 5).
  end for
  $F(step) = \sum_{i=1}^{N} \|\vec{F}_i\|$.
  $\psi(step)$ is calcluated using Eq.( 8).
  $D_c(step)$ is calcluated using Eq.( 18).
end for
return  Force $F$, Alignment $\psi$ and Cohesiveness $D_c$.
```

effectively needs to make a well-organised structure which includes the particles' population size, parameters' limits, particles' velocity evaluation rule, etc. That structure can affect the optimisation result substantially. Therefore, it is necessary to organise the structure according to the aiming optimisation problem. If the PSO structure is adjusted properly, each particle systematically seeks its personal best position (P_{best}) based on its individual experience and the global best position (G_{best}), dictated by the collective experience of its neighbour particles, as illustrated in Figure 4.

The i^{th} particle is composed of three vectors: (i) the x-vector that stores the current position, (ii) the p-vector that stores the best position found by the particle, and (iii) the v-vector that contains the velocity of the particle. The following equations are used to calculate the position x and velocity v of the i^{th} particle:

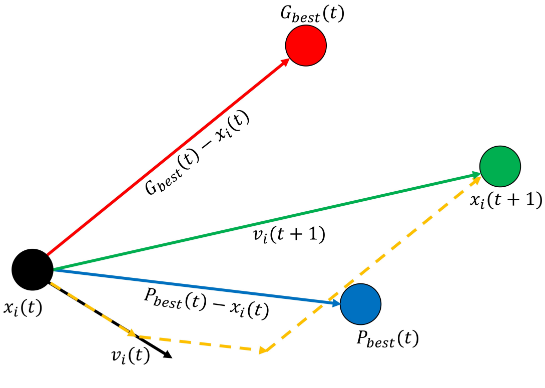

Figure 4: The movement of one particle in the search space of the PSO algorithm.

$$v^i_{t+1} = w\, v^i_t + c_1\, r_1\, (P_{best} - x^i_t) \; + \; c_2\, r_2\, (G_{best} - x^i_t) \; , \tag{9}$$

$$x^i_{t+1} = x^i_t \; + \; v^i_{t+1} \; . \tag{10}$$

Here, $t = \{1, 2, 3, ..., I_{max}\}$ corresponds to the iteration count in which I_{max} denotes the maximum number of iterations, and $i = \{1, 2, 3, ..., N\}$ signifies the individual particles with N marking the overall size of the particle swarm. r_1 and r_2 are random variables ranging between 0 and 1, and w denotes the weight coefficient. c_1 is the cognitive scaling parameter affiliated with the optimal local position P_{best}, while c_2 represents the social scaling parameter associated with the optimal global position G_{best}. Despite the simplicity of the PSO methodology, premature convergence emerges as a notable pitfall in the exploration process of the PSO algorithm. This effect essentially impedes the capability of the algorithm to thoroughly explore the solution space. Accordingly, specific strategies will be adopted in this study to enable particles to escape from the local minimum traps. Clerc and Kennedy [37] introduced the concept of constriction coefficients to optimally determine the values of w, c_1, and c_2. This approach seeks to maintain an equilibrium between local and global exploration and is expressed through the subsequent equations:

$$\chi = \frac{2\, k_c}{|2 - \phi - \sqrt{\phi^2 \; - \; 4\,\phi}|} \; , \tag{11}$$

$$\phi = \phi_1 + \phi_2 \; . \tag{12}$$

According to Eq. (11), the value of k_c can be selected between 0 and 1. The values of ϕ_1 and ϕ_2 can be adjusted in which their sum is higher than four ($\phi > 4$). After preparing the necessary coefficients, the parameters w_{max}, c_1 and c_2 can be calculated as follows:

$$w_{max} = \chi \; , \; c_1 = \chi\, \phi_1 \; , \; c_2 = \chi\, \phi_2 \; . \tag{13}$$

Furthermore, an adaptive inertial weight technique [38] will be integrated to increase the probability of circumventing the risk of local minima trap for the PSO algorithm. The strategy initiates with a large value of w_{max} by calculating its value using Eq. (13). Typically, the range of $w_{max} \in [0.9, 1.2]$ is used to promote (exploration) in the solution space. Subsequently, this value is iteratively reduced until it attains $w_{min} \in [0.4, 0.5]$ to refine solutions (exploitation). The adaptive inertial weight can be calculated as follows:

$$w = w_{max} - step\; \frac{(w_{max} - w_{min})}{I_{max}} \; , \tag{14}$$

where, w_{max} represents the maximum value of the inertia weight and w_{min} denote the minimum value of the inertia weight. *step* is the current iteration number in the loop, and I_{max} is the maximum number of iterations in the PSO algorithm.

Within the scope of this study, the implementation of the PSO algorithm incorporates two termination conditions. The first condition is associated with the iteration count reaching its maximum permitted value, which is frequently encountered in numerous research studies. The second condition relates to the convergence of the particles toward the optimal solution. This is achieved by scrutinising the coefficient of variation (CV) of the most recent optimal cost values computed. During the execution of the PSO algorithm, the absolute difference between the mean of P_{best} values of all particles and the global best G_{best} is archived in an array denoted as (GBC). The CV is characterised as the ratio of the standard deviation to the mean. In the context of the PSO algorithm, if the CV of the GBC over the last 20 iterations falls below a threshold (< 0.05), it signifies a stagnation in the improvement of the particle swarm, thereby implying probable convergence towards the optimal solution. This termination criterion enhances efficiency by enabling early termination of the algorithm if significant progress is no longer discernible, thereby optimising computational time.

4.2. Cost Function

The definition of an appropriate cost function, also known as a fitness function, is a critical step in the optimisation procedure. In this paper, a multi-objective cost function is proposed which includes the vital characteristics of a swarm robotic system. The OCM model is optimised by minimising three main objectives: i) the alignment error ψ_{error} of the swarm, i) the time required for alignment convergence, and iii) the virtual forces invoked by the individuals of the swarm. The formula of the cost function is as follows:

$$J = \sum_{t=0}^{T_{max}} \left[w_1\ \psi_{error} + w_2\ t_{rise}^2\ + w_3 \sum_{i=1}^{N} F_i \right]. \tag{15}$$

In Eq. (15), N denotes the size of the swarm incorporated in the OCM model, and T_{max} represents the maximum permissible simulation time. One of the objectives is to increase the alignment ψ to its maximum value of '1', signifying the robots are moving in near-identical directions. However, the designed cost function aims to minimise and therefore the alignment error which is computed as $\psi_{error} = (1 - \psi)^2$ is used in the objective function to ensure that ψ_{error} reaches the minimal possible value. t_{rise} indicates the time at which 60% of the robots have achieved alignment. $\vec{F_i}$ denotes the induced force on the i^{th} robot, generated by the viscoelastic links. Moreover, three weighting parameters, w_1, w_2, and w_3, are introduced into the cost function as tuning parameters to adjust the importance level for each part of the objective function. These weights allow for the balancing of objectives in the multi-objective optimisation function, providing flexibility in emphasising certain objectives over others based on specific requirements or constraints.

5. Implementation and Experiment

This section presents the experimental structure of the OCM model used in the conducted research. The experiments were designed to manifest the evolution of OCM performance when compared with two preceding studies [23, 12]. The system's robustness against different noise levels has been investigated in this research as well. Two categories of experiments have been conducted: i) numerical simulations and ii) real-world robotic experiments.

5.1. Numerical Simulation

5.1.1. Simulation Setup

The numerical simulations have been executed on MATLAB, aiming to study the feasibility and performance of the proposed OCM model. A Mobile Robotics Simulation Toolbox was utilised to perform the OCM model simulations. In these simulations, each robot is graphically represented as a blue circle (denoting the body) and a small black line (indicating the heading), as visualised in Figure 3. The robots are prepositioned at specific locations, with random orientations within a two-dimensional arena of size $L \times L$. It should be noted that all conducted simulations maintain a square configuration for the swarm shape. Adhering to the Nearest-neighbour topology for the OCM model experiments, all robots are fully interconnected with their adjacent robots, as illustrated in Figure 3. The longest link between two adjacent robots is the diagonal link, which has been defined as the sensing radius R_s. At the outset of the experiments, the natural length l_{ij} is computed using the Euclidean distance formula, taking into account the initial inter-robot distances. At $t = 0$ s, no force generation occurs amongst the robots. However, as the experiment progresses, various attractive and repulsive forces are invoked due to the dynamic behaviour of the OCM model.

The Monte Carlo method has been implemented by conducting 80 simulations to ensure the proposed system's validity with the presence of measurement noise. The primary parameters of the experiments, such as the maximum simulation time $T_{max} = 300$ s, swarm size $N = 100$ robots, arena size $L = 100$ m, and sensing range $R_s = 7.1$ m, were initialized. A constant robot velocity of $v_0 = 0.075$ m/s was set. A comprehensive list of the utilized variables in the experiments is provided in Table 1.

5.1.2. PSO and Cost Function

The PSO algorithm is employed to optimise the control parameters of the OCM model: α, β, k, and c. Several parameters are required to ensure the PSO algorithm's efficient operation: the maximum iteration number is set as $I_{max} = 100$, and the particle size is designated as $n_{pop} = 200$. Furthermore, maximum and minimum bounds are established for the particles' velocities and decision variables to ensure that the i^{th} particle values are contained within a prescribed range. Guided by Eq.(11) and Eq.(12), the parameters are set as $k_c = 1$, $\phi_1 = \phi_2 = 2.05$, which aid in determining appropriate values for w_{max}, c_1, and c_2 for use in Eq. (9) and Eq. (14).

The cost function considers three input variables: the alignment degree, convergence time, and virtual force. Each particle in the PSO algorithm implements the cost function at every position within the search space. Subsequently, the PSO algorithm assesses the output values of the cost function, selecting the minimum value as the optimal solution. It is worth highlighting that the optimisation process is conducted in the absence of noise, reflecting an idealised environment for the implementation of the OCM model. The weight parameters w_1, w_2, and w_3 in the cost function, as illustrated in Eq. (15), are empirically tuned to ensure a uniform data range across all three terms.

5.1.3. *Noise and Disturbance*

The OCM model considers two types of noise: (i) D_r, which is associated with positions, and (ii) D_θ, associated with orientations, as elaborated in Eq.(4) and Eq.(5). Since both noises originate almost similar outputs [39], the same noise levels are assigned for both D_r and D_θ. Four noise intensity levels, denoted as η, are utilised to examine the OCM model's performance, with $\eta \in \{0, 0.1, 0.2, 0.3\}$. Note that $\eta = 0$ is used to assess the performance of the proposed model in the absence of noise. In this paper, we assume that the last level $\eta = 0.3$ is considered the highest noise value. The process of introducing noise into the system involves initially selecting the noise intensity η. Then, at each iteration, two uniformly distributed random values within the range $[0, \eta]$ are generated and assigned to both D_r and D_θ.

Table 1: List of simulation parameters used in this paper.

Parameters	Description	Values
N	Number of robots	100
v_0	Constant speed	0.075 [m/s]
L	Side length of the arena	100 [m]
R_s	Sensing radius of a robot	7.1 [m]
T_{sim}	Time of a simulation run	300 [s]
n_{pop}	Number of particles in PSO	200 particles
I_{max}	Max iteration of PSO	100
w_{min}	Min value of inertia weight	0.4
w_{max}	Max value of inertia weight	1
ϕ_1, ϕ_2	Constriction coefficients	2.05
w_1	Weight for alignment degree	0.006
w_2	Weight for convergence time	6
w_3	Weight for virtual force	0.07
η	Noise levels	{0,0.1,0.2,0.3}

5.2. *Real-world robots experiments*

5.2.1. *Swarm Robotics Platform: MONA*

The open-source miniature robot MONA, as depicted in Figure 5, was developed for effective robotics application [40]. It serves as an affordable, user-friendly mobile robot for both educational and research purposes. The robot has a System-on-a-Chip (SoC) ESP32-WROVER-B, operating at up to 240 MHz with a 4 MB flash memory and 8 MB PSRAM. The motion of the robot is powered by two DC motors, providing a top speed of 1.5 cm/s. For proximity detection, MONA is fitted with five evenly distributed infrared (IR) sensors on its front, each spaced 35° apart. Furthermore, MONA hosts a Wi-Fi module facilitating wireless communication with other robots in the swarm. The onboard lithium battery offers an operational time between 1.2 and 2.3 hours, with battery status continuously monitored via an inbuilt power management module. Each motor, independently controlled by an H-bridge DC motor driver, consumes electrical power in the range between 100 mW and 200 mW, contingent on the load and speed.

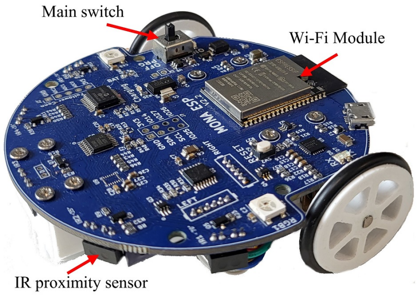

Figure 5: The MONA mobile robot utilised in the real-world experiments.

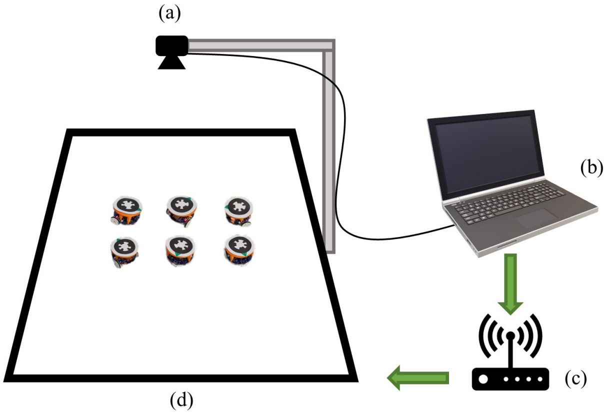

Figure 6: Setup of the experiment displaying the main components: (a) camera, (b) master PC, (c) wireless router, and (d) Arena.

5.2.2. *Experimental Arena Configuration*

In order to evaluate the applicability of the proposed collective motion model, an experiment is designed wherein six robots are located within a 2 × 1 m^2 arena. For all experimental runs, robots are positioned at predefined positions with randomly oriented angles, ensuring that each trial begins from a disordered state. A high-resolution camera is installed approximately two meters above the arena and connected to a master PC to oversee the robot swarm. The robots' movements during the experiment are tracked via the WhyCon vision-based localisation system [41]. This system provides precise estimations of the robots' positions using a low-cost webcam. Distinctive fiducial circular tags are placed on the top of each robot, enabling the WhyCon system to identify each robot's position, orientation, and ID, as presented in Figure 6.

As depicted in Figure 6(a), the camera captures the positions and orientations of all swarm individuals, which are then transmitted via Robotic Operating System (ROS) communication to the OCM flocking controller in the master PC. In Figure 6(b), each i^{th} robot computes the force based on its neighbours' information, leading to the determination of new positions, $\vec{x}_i$, and orientations, $\dot{\theta}_i$. By implementing a simple p-controller, these variables are converted into right and left motor speeds, $v_r = V + k_p\ W$ and $v_l = V - k_p\ W$, respectively. The linear velocity V is defined as a constant value $V = 100$ m/s, representing the minimum value that ensures the DC motors of all the robots can move. w denotes the angular velocity that is defined as $W = \dot{\theta}_i$ which represents the heading error. k_p is the p-controller coefficient. In Figure 6(c), the calculated left and right speeds are transmitted to the MONA robots through a wireless router. Each MONA robot is equipped with an ESP32 WiFi module to receive data from the master PC via the wireless router. Finally, Figure 6(d) provides a view of the arena where all robots are placed in a predefined formation and randomly oriented. Additionally, the fiducial circular tags can be seen atop each robot.

Every experiment includes a set number of iterations to achieve the desired flocking behaviour. During each iteration, the OCM algorithm generates a new set $[x, y, \theta]^T$ for each robot. Sequentially, each MONA robot updates its position and angle based on these values. The updating of the last robot means the completion of one iteration, after which the OCM algorithm is computed again to obtain the next set of positions and angles. Each iteration, on average, lasts for 7 s, resulting in a total approximate experiment duration of 392 s. Two criteria are established to terminate the experiment. The experiment concludes when either of these conditions is met: the first condition is reaching the maximum number of steps, denoted as $sim\_num = 56$, and the second condition is activated when any robot exceeds the specified boundaries of the arena. It is crucial to note that certain iterations may not require all robots to update their $[x, y, \theta]^T$ information. This could happen when part of the swarm attempts to steer to the new heading while the rest of the robots are in the correct heading. Therefore, the time required to update the positions and orientations of each iteration is not constant throughout the experiment.

Although the real-world experiments conducted in this research could theoretically employ a decentralised approach, where the relative positions are derived from the onboard sensors, a centralised methodology was deliberately chosen. This decision was substantially influenced by the specific attributes of the MONA robot. Given that MONA features relatively limited sensing capabilities, the accuracy of the data obtained from its immediate environment could be potentially compromised if a decentralised method were adopted. Consequently, the choice of a centralised approach serves to mitigate these inaccuracies and ensure the robustness of the experimental data.

5.3. Performance Metrics

The OCM model is applied to performing a flocking behaviour. Two significant factors are defined: i) degree of alignment ψ, and ii) swarm's cohesiveness D_c to evaluate the collective motion behaviour of the swarm. The Degree of Alignment ψ demonstrates how agents are aligning closely, thereby reflecting the overall directionality of the swarm. This factor is extensively analysed in the next sections by implementing Eq. (8).

On the other hand, the swarm's cohesiveness D_c estimates how well the swarm maintains the consistency of its shape. Cohesiveness can be determined based on the distance between the robots and it is commonly employed in several related studies [42] [43] [44]. In this section, the distances between the robots and the swarm's centre of mass are obtained. Then, the coefficient of variation is calculated by dividing the standard deviation of the distances by the average of the distances. This is mathematically expressed in the following equation:

$$\mu_c = \frac{\sum_{i=1}^{N} \|r_i - r_c\|}{N} \tag{16}$$

$$\sigma_c = \sqrt{\frac{\sum_{i=1}^{N} (\|r_i - r_c\| - \mu_c)^2}{N - 1}} \tag{17}$$

$$D_c = \frac{\sigma_c}{\mu_c} \tag{18}$$

Here, N represents the number of robots in the swarm. r_c and r_i signifies the centre position of the swarm and the position of the i^{th} robot, respectively. $\|r_i - r_c\|$ to find the Euclidean distances from each robot to the centre of mass. μ_c is the average distance of the robots from the centre, and σ_c denotes the standard deviation of the Euclidean distances. D_c is the ratio of the standard deviation of the Euclidean distances to the average distance, which represents the coefficient of variation. In order to ensure that the overall swarm preserves a stable form throughout the experiment, it is essential that the value of D_c remains consistent during the simulation run. Fluctuations in the value of D_c imply lower cohesiveness, and the swarm is more dispersed.

6. Results

6.1. Numerical Results

A comprehensive set of experiments was carried out to investigate the performance of the proposed OCM model utilising the PSO algorithm. In this research, we took into account three primary objectives: (i) reduction of the resultant force exerted by the interaction of the robots, (ii) expedited swarm convergence, and (iii) maximisation of the agreement amongst robots' angles towards a common direction. The results presented in this section correspond to a swarm size of $N = 100$ robots. As mentioned previously, the OCM model was subjected to various levels of noise. However, for the purpose of performance evaluation, two noise levels are applied: i) without noise ($\eta = 0$) and ii) high noise ($\eta = 0.3$). The effects of other noise levels will be explored in the discussion section.

The optimised control parameters for the OCM model were determined as follows: $\alpha = 0.0262$, $\beta = 0.5627$, $k = 1.0$, and $c = 1.7503$. The PSO algorithm seeks the optimal solution until one of the termination conditions is met. The average value of running the cost function for 150 simulations to examine the optimisation accuracy and demonstrate the progressive improvement of the PSO algorithm is depicted in Figure 7. This optimisation result was achieved with 200 particles in the search domain. The shaded plot reveals that the optimisation of each simulation tends to converge approximately after 60 iterations. Moreover, the bounds around the average line remain narrow

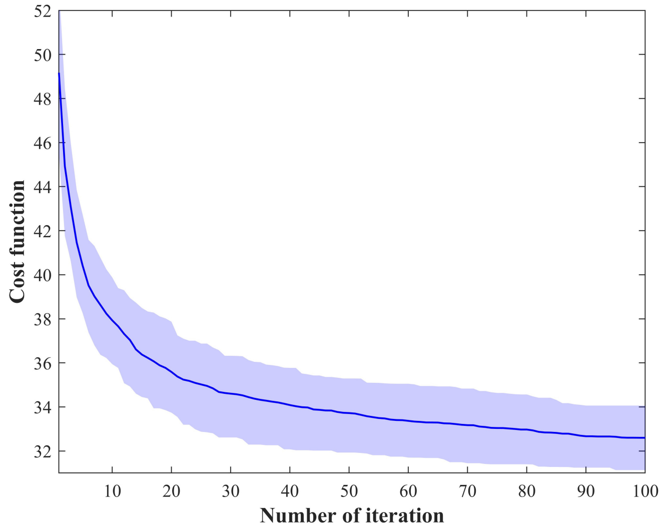

Figure 7: Monte-Carlo simulations of PSO algorithm repeated 150 times.

Table 2: The values of the control parameters that are used for comparison in this work.

Model	α	β	k	V_0	c
Original [23]	0.01	0.12	5	0.075	-
AESPSO [12]	0.18082	0.81649	1.0	0.075	-
OCM	0.0262	0.5627	1.0	0.075	1.7503

through the maximum iterations, which demonstrates the robustness of the optimisation algorithm used in this study. This evidence highlights the critical role of the PSO in bolstering collective motion behaviour. The weight coefficients w_1, w_2, and w_3 were empirically tuned to produce satisfactory outputs in terms of cohesiveness, quick swarm alignment convergence, and minimum control effort.

The proposed OCM algorithm utilising the optimised control parameters in this paper has exhibited a significant improvement in the collective motion of the swarm compared to the results obtained in previous works presented in [23, 12]. Table 2 displays the control values of the OCM model and the collective motion frameworks used to examine the performance of the proposed work. It is noteworthy that only the OCM model utilises a damper with an optimised value of the corresponding parameter c.

In order to evaluate the OCM algorithm's performance, four cases are examined: (1) OCM-NN, (2) OCM-HN, (3) AESPSO-HN, and (4) original-HN. (NN) implies that the implementation was executed without noise $\eta = 0$, while (HN) suggests that a high noise value $\eta = 0.3$ was used. Figure 8 illustrates four instances at different time intervals $t = \{0, 50, 150, 300\}$ s of the four cases. Note that these instances were randomly selected from 80 simulations. Moreover, the constant speed v_0 of each individual in the swarm is unified across all four cases to ensure a fair comparison, as reflected in Table2.

Figure 8(a) reveals that a high level of noise, $\eta = 0.3$, was used with the original AES algorithm. The robots exhibit suboptimal collective motion behaviour throughout the simulation time, struggling within the arena. Despite the robots' attempts to maintain proximity, the shape of the swarm is deformed due to the high values of attraction and repulsive forces, leading to a low degree of alignment, particularly visible at $t = 300$ s. In Figure 8(b), a high level of noise, $\eta = 0.3$ is applied in conjunction with AESPSO parameters from previous work. Fluctuations due to the induced forces among the robots are observable at $t = 50$ s. By $t = 150$ s, the robots are striving to align and counteract the noise presence, causing a slight deformation in the swarm shape. However, by $t = 300$ s, approximately 90% of the swarm's individuals have aligned. Despite AESPSO control parameters demonstrating acceptable performance in the presence of noise, the convergence time for swarm alignment is slow. Furthermore, the degree of alignment in the AESPSO-HN case is not consistent across the 80 simulations, as can be seen in Figure 10.

Table 3: The median values of the force and alignment of the four investigated scenarios at $t = 300$ s.

Scenario	Force (F)	Alignment (ψ)
Original-HN	138.41	0.390
AESPSO-HN	14.49	0.705
OCM-HN	8.25	0.996
OCM-NN	4.75	0.999

In Figure 8(c), the proposed OCM model is introduced in the presence of high noise levels, $\eta = 0.3$, to assess system robustness. Interestingly, by $t = 50$ s, almost the entire swarm exhibits alignment, signifying rapid swarm convergence. The degree of alignment remains stable throughout the simulation. It should be noted that although the robots maintain an ordered state, the swarm frequently changes direction due to the noise, indicating a versatile change in the collective motion direction. Lastly, Figure 8(d) represents the implementation of the OCM model under ideal conditions, without noise $\eta = 0$. Here, the rapid convergence and stability of the swarm are quite evident. Furthermore, the swarm maintains a consistent direction throughout the simulation. The OCM algorithm with optimised parameters demonstrates similar performance in the two last cases. Accordingly, it is evident that the proposed framework establishes considerable robustness in the presence of measurement noise. Furthermore, comparing the two last cases with the former algorithms' performance substantiates a notable improvement regarding the rise time for the alignment.

The Monte Carlo simulation is exploited to investigate the robust performance of the algorithm in comparison with former methods in more detail. In each of Figures 9 and 10, four shaded plots represent the four previously mentioned cases. In these plots, the middle line denotes the median value, while the upper and lower lines signify the third quartile and first quartile, respectively. The black plot signifies the original-HN, the green plot represents AESPSO-HN, the red plot represents OCM-HN, and the blue plot depicts OCM-NN. Table 3 shows the median values of the four investigated cases at $t = 300$ s to observe the collective motion performance based on Figures 9 and 10.

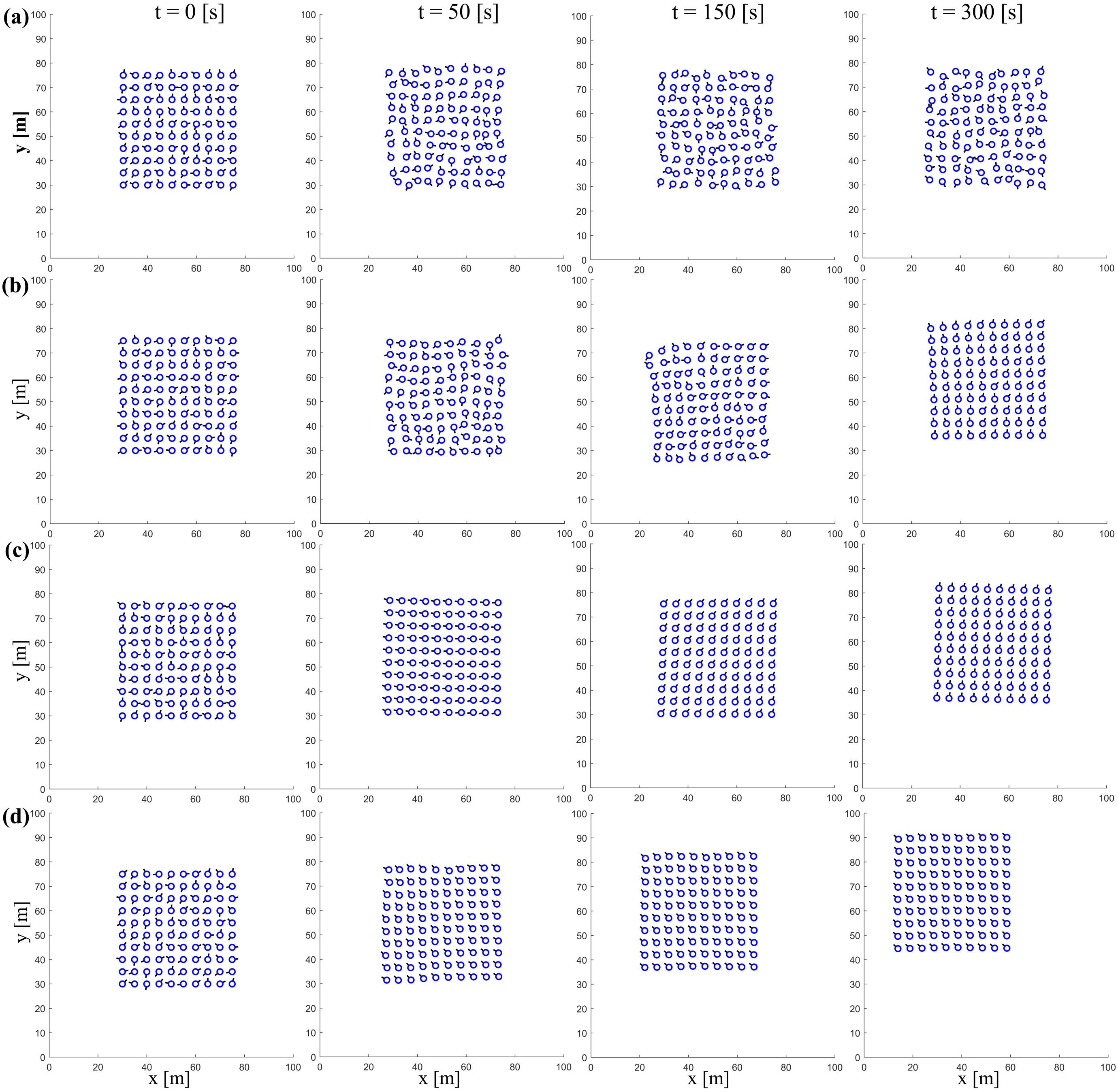

Figure 8: Simulations of $N = 100$ robots where each row represents one of the four different cases: (a) Original-HN, (b) AESPSO-HN, (c) OCM-HN, and (d) OCM-NN. Snapshots are taken for each case at four different time intervals $t = \{0, 50, 150, 300\}$ s.

Upon examination of the two figures, it is clear that the original AES performance is corrupted by the measurement noise, and although the system remains bounded and stable, the final result is far from the optimal solution for both control effort and alignment. Those results imply that the application of the original AES parameters in noisy conditions leads to outcomes far from the optimal point, suggesting that a re-evaluation or adjustment of these parameters could be beneficial in improving swarm behaviour.

Figure 9 presents the results for the average force vector magnitude within the swarm as a measure of control effort required to keep the swarm stable and cohesive. The AESPSO-HN demonstrates a satisfactory performance, as evidenced by a reasonably narrow variation bound. Furthermore, the OCM-HN performance is quite close to the result obtained for the OCM-NN which means the control effort required for stabilisation and cohesiveness doesn't increase significantly to regulate the effect of noise.

Figure 10 reflects the comparison results for the alignment of the four investigated cases. According to the presented re-

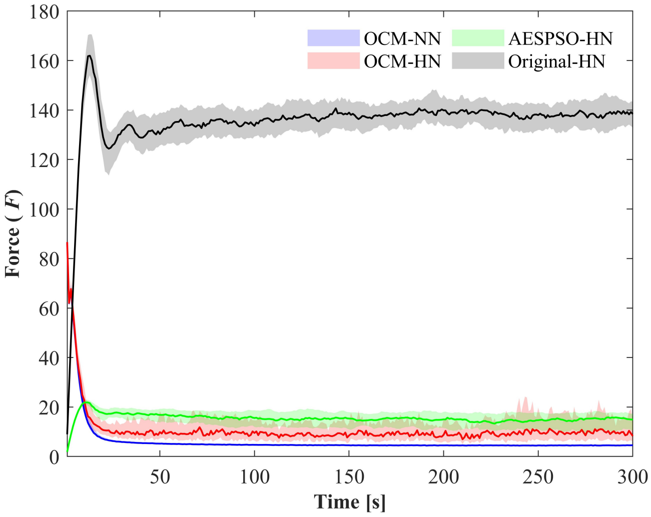

Figure 9: Total virtual force of N=100 robots for the four designed cases repeated 80 times.

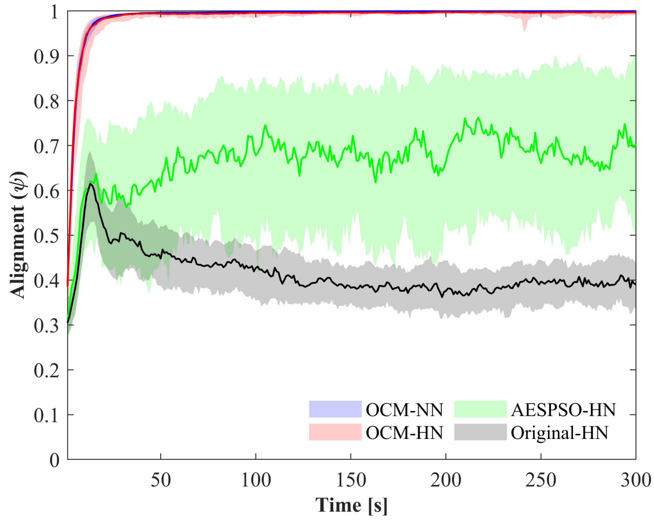

Figure 10: Degree of alignment of N=100 robots for the four designed cases repeated 80 times.

sults, the original AES cannot stand against the noise in the system, and the alignment even decreases after an initial rise, although it remains bounded. The AESPSO-HN shows a more appropriate alignment regarding the steady increase in the median line. However, the variation bound is substantially wide with respect to the median signal. It clearly signifies that the AESPSO doesn't show a reasonably robust performance for the alignment. Moreover, that algorithm couldn't have reached an alignment better than 90% in all simulations. On the other hand, the proposed OCM algorithm demonstrates remarkable performance in alignment with or without noise. The variation bound for the alignment is quite small with respect to the median line, which means the proposed framework keeps all the agents aligned regardless of the noise in the system with minimum control effort. Furthermore, the swarm reaches almost complete alignment, $\psi \simeq 1$, outperforming former methods substantially. Although alignment can establish how the algorithm can make the agents move in the same direction, it cannot show if the algorithm can keep the agents in a structure effectively. Therefore, cohesiveness is another criterion that evaluates the collective motion performance and demonstrates the algorithm's capability to maintain the swarm structure uniform. Figure 11 presents the cohesiveness results for all the cases investigated in this paper. The obtained results are consistent with the results obtained for the alignment, which means that the proposed OCM framework outperforms former algorithms with respect to cohesiveness as well. Furthermore, the narrow variation bound establishes the robustness in keeping the swarm shape regardless of measurement noise presence.

6.2. Hardware Results

Hardware experiments serve as an essential validation step for the performance of the Optimised Collective Motion (OCM) model, utilising a group of six MONA robots. Throughout the experiments, the robots maintain full connectivity via the Nearest-neighbour network. Similar methodologies and theoretical perspectives have been previously addressed in [25]. As mentioned earlier, two pre-defined conditions to terminate the hardware experiment: (1) the instance wherein a MONA robot exceeds the boundary of the arena, and (2) the moment the experiment reaches the predetermined maximum number of steps, denoted as sim_num = 56. Figure 12 illustrates four snapshots of the OCM experiments at four distinct moments: (a) t = 0 s, (b) t = 50 s, (c) t = 150 s, and (d) t = 250 s. The heading of each robot is determined with a small red triangle for clarity. Figure 12(a) shows the initial positions and random orientations of six MONA robots at t = 0 s, at which the swarm is in a disordered state. Each robot during the experiment received the speeds of the right and left motors via wireless communication. MONA robots struggled to align and maintain the rectangle shape because of the high attractive and repulsive forces at t = 50 s as evidenced by Figure 12(b). Moreover, a minor perturbation is observed due to the noise effect from the localisation system. Interestingly, at t = 150 s, the swarm's internal fluctuations are decreased, and the robots attempt to share the same direction as presented in Figure 12(C). From t = 250 s onwards, as shown in Figure 12(d), the robots are approximately

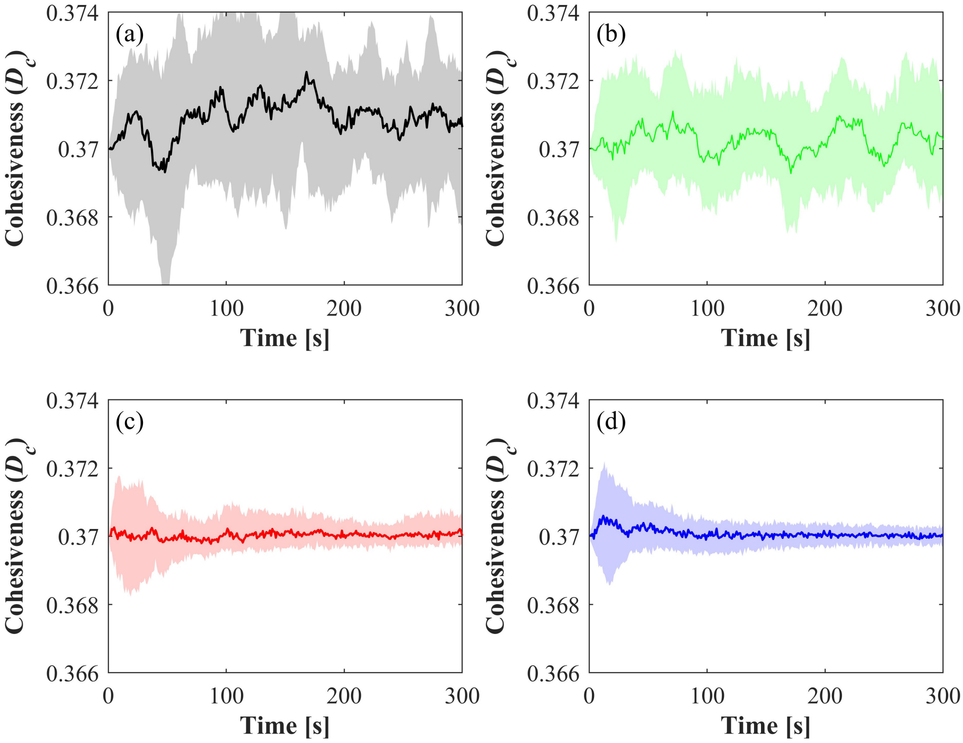

Figure 11: The cohesiveness of N=100 robots for the four designed cases: (a) Original-HN, (b) AESPSO-HN, (c) OCM-HN, and (d) OCM-NN.

fully aligned, and the rectangle shape of the whole swarm is adjusted, which indicates the improvement in the order state of the swarm. Furthermore, it is evident that the swarm shifts slowly towards the left side of the arena. The direction of the swarm's movement is intrinsically random, resulting from the interplay of induced virtual forces and the swarm's consensus on heading. The successful achievement of collective motion thereby validates the feasibility and practicality of the proposed OCM model in real-world experiments.

Figure 13 presents the degree of alignment derived from experimental results obtained through the repetition of the experiment 27 times, as depicted in Figure 12(a) - (d). From an analysis of Figure 13, it can be observed that all the robots initially showed a state of disorder in their orientations across the repeated experiments, approximately $\psi = 0.149$. The broad shaded area across the period $t = 60$ s to $t = 190$ s signifies the hard efforts of the robots to align and attain stability in their movement trajectories. An analysis of the quantitative data from these 27 experiments indicates that the robots struggled to be aligned in approximately 20% of the experiments. From $t = 200$ s onwards in Figure13, both the rise in the median of the alignment and the reduction in the shaded area are noticeable. These graphical representations suggest a satisfactory convergence towards alignment that is consistent with the outcomes derived from simulation experiments.

7. Discussion

This study proposes an Optimised Collective Motion (OCM) model, which fundamentally incorporates viscoelastic links among the robots. These viscoelastic connections, comprised of spring-damper models, are designed to mitigate fluctuations between robots, thereby minimising the control effort. The proposed OCM framework establishes robust performance in the presence of noise, considering both numerical simulations and real-robot experiments. To gain a deeper understanding of swarm performance using the OCM model, an extensive series of tests were conducted to explore the influence of various critical factors. These factors included the noise intensity (η), the constant velocity (v_0), the size of the swarm (N), the presence or absence of the damper, and the swarm shape. The collective motion behaviour is evaluated according to the variations applied to these factors, considering the alignment as the performance criteria. A Monte Carlo method, executing 50 simulations, is employed to examine the robustness of the system. All the experiments in the following sections were conducted by applying the highest noise value in this paper ($\eta = 0.3$) except the tests in Section 7.1. The statistical results are presented with box plots, which include mean value, first and third quartiles, and exclusive variation bounds. The median values from these boxplots are utilised in subsequent sections to compare the performance of the OCM model under different conditions. In addition, the y-axis of the alignment metric (ψ) is adjusted based on the results to visually clarify the performance.

7.1. *Impact of noise intensity (η)*

A series of tests were conducted to assess the robustness of the Optimised Collective Motion (OCM) model in relation to variations in measurement noise intensity. As mentioned in

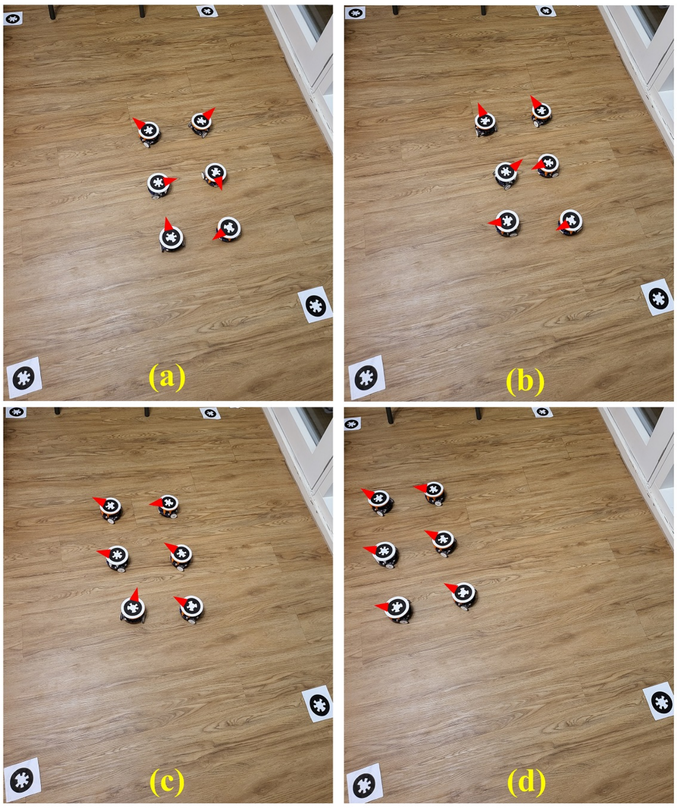

Figure 12: Hardware experiment of six MONA robots on 2 × 1 m^2 arena. A localisation system is used to detect the positions and angles of the robots. (a) Initial position and random orientations at $t = 0$ s; (b) Robots attempt to align at $t = 50$ s; (c) The robots gradually share the desired angles at $t = 150$ s, and (d) At $t = 250$ s onwards, the robots reach the consensus level of alignment.

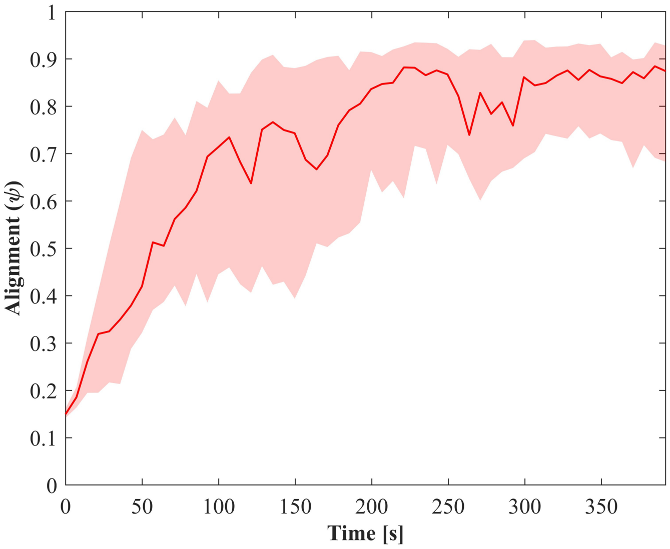

Figure 13: Degree of alignment for a swarm of six MONA robots through 27 experimental runs. The duration of each experiment is from $t = 0$ s to $t = 392$ s. The middle line denotes the median value. The upper and lower lines represent the third and first quartiles, respectively.

Section 5.1.3, diverse noise values are implemented, specifically $\eta \in \{0, 0.1, 0.2, 0.3\}$, to observe their respective impacts on the behaviour of the swarm. Throughout these tests, the size of the swarm was maintained at $N = 100$ robots, the constant velocity was set at $v_0 = 0.075$ m/s, and the remaining parameters were kept fixed as well. Despite being subjected to a range of noise intensities, the performance of the OCM model exhibited a degree of resilience, with the variations in performance being gradual and not significantly divergent, as is evident in Figure 14. It is distinctly noticeable that at a noise intensity of $\eta = 0$, the system delivers optimal alignment, achieving $\psi = 0.9888$. However, as the noise intensity increases, the degree of alignment experiences a marginal decline, reaching $\psi = 0.9868$, $\psi = 0.9706$, and $\psi = 0.9564$ for $\eta = 0.1$, $\eta = 0.2$, and $\eta = 0.3$. These findings substantiate the robustness of the OCM model in the face of increasing noise intensity, highlighting its practical applicability in environments where noise variation is a considerable factor.

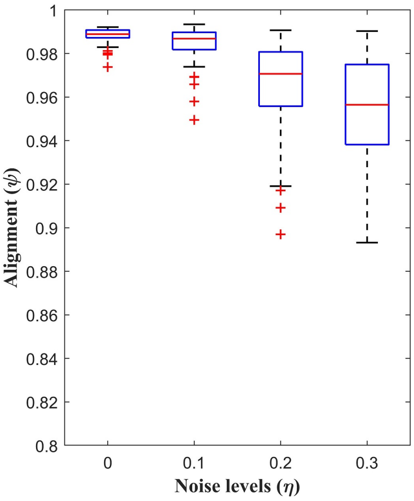

Figure 14: The degree of alignment of different noise intensity values.

7.2. *Impact of constant velocity (v_0)*

An analysis was conducted using various constant velocity (v_0) values in Eq. (4) to study its influence on the speed of alignment convergence. The simulations were carried out for $v_0 \in \{0.002, 0.01, 0.075, 0.2\}$ m/s, while keeping the other principal parameters fixed. The results of the simulations demonstrated that the highest velocity value of $v_0 = 0.2$ m/s exhibited the worst performance alignment, with the degree of alignment being $\psi = 0.31$. This can be attributed to the fact that a high velocity value resulted in an increase in fluctuations among the robots, thereby adversely affecting the stability of the swarm. However, the other three constant velocity values demonstrated a progressive improvement in performance, as depicted in Figure 15. The lowest velocity $v_0 = 0.002$ m/s makes the swarm converge slowly. The next increment in velocity, $v_0 = 0.01$ m/s, displays a notable increase in the degree of alignment, achieving $\psi = 0.88$. This indicates that the swarm can reach a consensus on their directional orientation. Interestingly, $v_0 = 0.075$ m/s, which is the selected velocity value in this paper, presented the most efficient performance with a high degree of alignment of

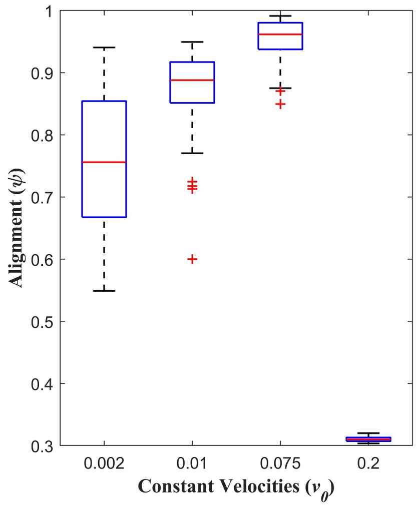

Figure 15: The degree of alignment for different velocities values.

ψ = 0.96. These results show the influence of the constant velocity parameter on the performance of the OCM framework in achieving efficient swarm behaviour.

7.3. *Impact of swarm size (N)*

A series of simulations were conducted with different swarm sizes $N \in \{100, 400, 900\}$ robots to investigate the influence of swarm size on the performance of the OCM framework. As illustrated in Figure 16(a), despite $N = 100$ robots demonstrating the optimal performance, the degree of alignment values for all swarm sizes are close, which emphasises the scalability of the algorithm. Upon observation of the median values of each boxplot, the alignment values are $\psi = 0.958$, $\psi = 0.944$, and $\psi = 0.941$ for swarm sizes $N = 100$, $N = 400$, and $N = 900$ respectively. This reveals a minor robot alignment decrease as the swarm size increases. These results clearly confirm the scalability feature of the OCM model and the stable performance in different swarm sizes.

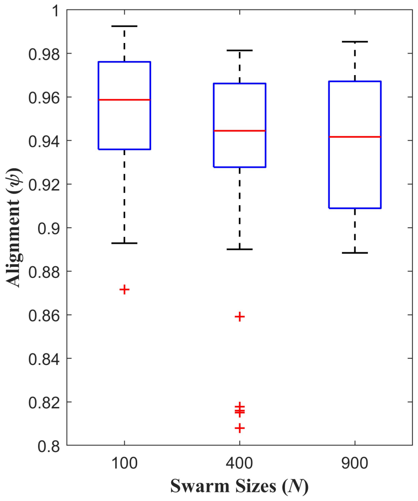

Figure 16: The degree of alignment for different swarm sizes.

7.4. *Impact of Damper*

The implementation of damper links between the robots serves to mitigate the fluctuations that arise due to the interplay of attractive and repulsive forces. In the context of a swarm size of $N = 100$ robots and a fixed velocity $v_0 = 0.075$m/s, two distinct experiments are performed: i) employing the optimised damper coefficient c, and ii) eliminating the damper coefficient where $c = 0$. The resultant influence of the damper on the degree of alignment is depicted in Figure17, with the results demonstrating a considerable enhancement when the damper is incorporated into the system, thereby emphasizing a significant contribution of this work. Figure 17 shows that integrating damper links between the robots has substantively demonstrated superior performance. The cohesiveness and stability of the swarm are significantly amplified, leading to a maximisation of the degree of alignment of the swarm, with $\psi = 0.965$. On the other hand, eliminating damper links causes a deterioration of the collective motion behaviour. The robots struggle to establish alignment throughout the simulation, with a substantially reduced alignment degree of $\psi = 0.416$.Therefore, as implemented in this study, the damper emerges as a crucial element in enhancing the performance of the OCM model.

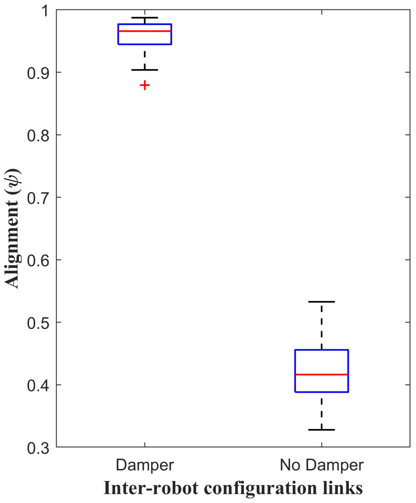

Figure 17: The degree of alignment with and without the presence of a damper.

7.5. *Impact of Swarm Shapes*

The Optimised Collective Motion (OCM) framework has demonstrated its robustness and stability in prior experiments, where robots were initially positioned in a square formation with random orientations. In this section, the performance of the OCM algorithm is investigated under two different initial configurations: i) a hexagonal shape and ii) a triangular shape. The parameters for the OCM are held constant across all tests, as shown in Table 1. According to Figure 18, it is noteworthy that the performance metrics associated with the collective motion across all distinct shapes are almost similar. The degrees of alignment are recorded as $\psi = 0.9841$, $\psi = 0.9845$, and $\psi = 0.9827$ for the square, hexagonal, and triangular formations, respectively.

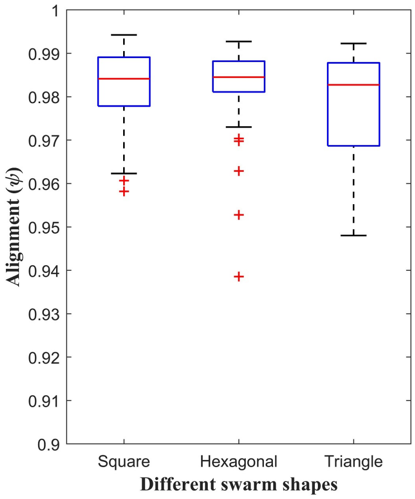

Figure 18: The degree of alignment of different swarm shapes.

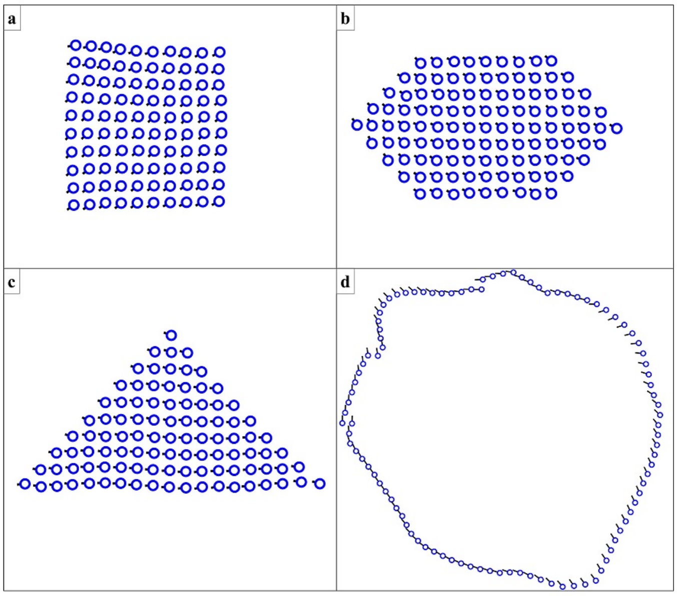

Figure 19: Snapshots of four different swarm shapes: a) square, b) hexagonal, c) triangle, and d) ring. The images of (a, b and c) are captured when 99% of the robots are aligned. Whereas the image of (d) is captured when 89% of the robots are aligned.

This observation is further clarified in Figure 19, which presents a snapshot of the three different shapes (a), (b), and (c) at a specific time. The capturing time is chosen to be the instance at which approximately 99% of the robots in the swarm are fully aligned to examine the rapid convergence. Upon visual inspection, it is apparent that the hexagonal shape demonstrates improved performance when compared to the square shape in terms of fast heading alignment. In Figure 19(a) and (b), 99% of the swarm in the square shape aligned at $t = 19$ s compared to $t = 13$ s in the case of the hexagonal shape. Conversely, the triangular shape exhibited much slower convergence with 99% of the swarm aligned at $t = 45$ s as shown in Figure 19(c). These findings demonstrate the influence of initial shape on the behaviour of the swarm and the OCM method.

Figure 19(d) depicts the movement of $N = 100$ robots in a ring shape where each robot is connected to only two nearest neighbours. The natural length $l_{ij} = 3.2$ and the rest of the parameters are fixed. The robots are placed in a circular configuration and randomly oriented. Initially, the robots attempted to align with their neighbours. Since each robot communicates with only two neighbours, the ring shape is divided into subgroups where each group points in a different direction. As shown in Figure 19(d), nearly 89% of the robots are aligned at $t = 90$ s. However, the swarm lost its stability, and the ring shape was deformed, as stated in [11]. This percentage value 89% of the aligned robots is the maximum value recorded throughout the repeated simulations.

8. Conclusion

This paper proposes a novel Optimised Collective Motion (OCM) control method by incorporating viscoelastic links between the robotic swarm individuals. These links substantially mitigate the undesirable swarm fluctuations among the robots, contributing significantly to fast alignment and preservation of the swarm's structure under measurement noise. The PSO algorithm was employed to fine-tune the control parameters of the proposed OCM algorithm. This optimisation process prioritised minimising virtual forces and alignment errors within the robotic swarm. The numerical results indicate a noteworthy improvement in the model's performance, particularly in terms of faster alignment and enhanced robustness in the presence of diverse noise intensities. Accordingly, the proposed method outperforms former algorithms significantly. Moreover, the impact of key factors, such as the robot's velocity, swarm size, swarm shape, presence of damper, and measurement noise, on the optimised model was extensively analysed, shedding light on the capabilities of the OCM control algorithm. Real-world experiments were conducted using a group of six mobile robots. Those experiments further confirmed the feasibility and effectiveness of the OCM model. While the proposed method showcases promising outcomes, it is essential to acknowledge its limitations. The algorithm is designed based on a particular neighbourhood strategy and it might increase the risk of isolation in large-scale workspaces. Furthermore, communication issues or other failures may cause unpredictable conditions resulting in instabilities. Future research directions could address these limitations and extend the algorithm's capabilities. Introducing additional virtual forces for navigation through complex and multi-target environments, extending the algorithm to cover three-dimensional applications, and obstacle avoidance would enhance swarm versatility.

Acknowledgements

This work was supported by EU H2020-FET RoboRoyale project (964492). Also, the first author would like to express special thanks to King Abdulaziz University (KAU) for supporting this work.

References

[1] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula, E. Bonabeau, Self-organization in biological systems, in: Self-Organization in Biological Systems, Princeton university press, 2020.

[2] C. K. Hemelrijk, H. Hildenbrandt, Schools of fish and flocks of birds: their shape and internal structure by self-organization, Interface focus 2 (6) (2012) 726–737.

[3] A. Mirzaeinia, F. Heppner, M. Hassanalian, An analytical study on leader and follower switching in v-shaped canada goose flocks for energy management purposes, Swarm Intelligence 14 (2020) 117–141.

[4] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, M. Sende, Swarm intelligence and cyber-physical systems: concepts, challenges and future trends, Swarm and Evolutionary Computation 60 (2021) 100762.

[5] D. Albani, J. IJsselmuiden, R. Haken, V. Trianni, Monitoring and mapping with robot swarms for agricultural applications, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2017, pp. 1–6.

[6] A. Din, M. Jabeen, K. Zia, A. Khalid, D. K. Saini, Behavior-based swarm robotic search and rescue using fuzzy controller, Computers & Electrical Engineering 70 (2018) 53–65.

[7] J. Hu, H. Niu, J. Carrasco, B. Lennox, F. Arvin, Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning, IEEE Transactions on Vehicular Technology 69 (12) (2020) 14413–14423.

[8] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, in: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 1987, pp. 25–34.

[9] T. Vicsek, A. Zafeiris, Collective motion, Physics reports 517 (3-4) (2012) 71–140.

[10] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, Physical review letters 75 (6) (1995) 1226.

[11] E. Ferrante, A. E. Turgut, M. Dorigo, C. Huepe, Collective motion dynamics of active solids and active crystals, New Journal of Physics 15 (9) (2013) 095011.

[12] M. Bahaidarah, F. R. Bana, A. E. Turgut, O. Marjanovic, F. Arvin, Optimization of a self-organized collective motion in a robotic swarm, in: International Conference on Swarm Intelligence, Springer, 2022, pp. 341–349.

[13] A. Cavagna, L. Del Castello, I. Giardina, T. Grigera, A. Jelic, S. Melillo, T. Mora, L. Parisi, E. Silvestri, M. Viale, et al., Flocking and turning: a new model for self-organized collective motion, Journal of Statistical Physics 158 (3) (2015) 601–627.

[14] H. Chaté, F. Ginelli, G. Grégoire, F. Raynaud, Collective motion of self-propelled particles interacting without cohesion, Physical Review E 77 (4) (2008) 046113.

[15] T. Ihle, Chapman–enskog expansion for the vicsek model of self-propelled particles, Journal of Statistical Mechanics: Theory and Experiment 2016 (8) (2016) 083205.

[16] F. Cucker, C. Huepe, Flocking with informed agents, Mathematics in Action 1 (1) (2008) 1–25.

[17] A. E. Turgut, H. Çelikkanat, F. Gökçe, E. Şahin, Self-organized flocking in mobile robot swarms, Swarm Intelligence 2 (2008) 97–120.

[18] A. M. Menzel, T. Ohta, Soft deformable self-propelled particles, EPL (Europhysics Letters) 99 (5) (2012) 58001.

[19] D. Strömbom, M. Siljestam, J. Park, D. J. Sumpter, The shape and dynamics of local attraction, The European Physical Journal Special Topics 224 (17) (2015) 3311–3323.

[20] D. Grossman, I. Aranson, E. B. Jacob, Emergence of agent swarm migration and vortex formation through inelastic collisions, New Journal of Physics 10 (2) (2008) 023036.

[21] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, M. Dorigo, Self-organized flocking with a mobile robot swarm: a novel motion control method, Adaptive Behavior 20 (6) (2012) 460–477.

[22] L. Barberis, F. Peruani, Large-scale patterns in a minimal cognitive flocking model: incidental leaders, nematic patterns, and aggregates, Physical review letters 117 (24) (2016) 248001.

[23] E. Ferrante, A. E. Turgut, M. Dorigo, C. Huepe, Elasticity-based mechanism for the collective motion of self-propelled particles with springlike interactions: a model system for natural and artificial swarms, Physical review letters 111 (26) (2013) 268302.

[24] A. E. Turgut, İ. C. Boz, İ. E. Okay, E. Ferrante, C. Huepe, Interaction network effects on position-and velocity-based models of collective motion, Journal of the Royal Society Interface 17 (169) (2020) 20200165.

[25] Y. Zheng, C. Huepe, Z. Han, Experimental capabilities and limitations of a position-based control algorithm for swarm robotics, Adaptive Behavior 30 (1) (2022) 19–35.

[26] M. Raoufi, A. E. Turgut, F. Arvin, Self-organized collective motion with a simulated real robot swarm, in: Annual Conference Towards Autonomous Robotic Systems, Springer, 2019, pp. 263–274.

[27] Z. Ban, J. Hu, B. Lennox, F. Arvin, Self-organised collision-free flocking mechanism in heterogeneous robot swarms, Mobile Networks and Applications 26 (6) (2021) 2461–2471.

[28] Z. Liu, A. E. Turgut, B. Lennox, F. Arvin, Self-organised flocking of robotic swarm in cluttered environments, in: Annual Conference Towards Autonomous Robotic Systems, Springer, 2021, pp. 126–135.

[29] Z. Ban, C. West, B. Lennox, F. Arvin, Self-organised flocking with simulated homogeneous robotic swarm, in: International Conference on Collaborative Computing: Networking, Applications and Worksharing, Springer, 2021, pp. 3–17.

[30] O. Michel, Cyberbotics ltd. webots™: professional mobile robot simulation, International Journal of Advanced Robotic Systems 1 (1) (2004) 5.

[31] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[32] S. Shao, Y. Peng, C. He, Y. Du, Efficient path planning for uav formation via comprehensively improved particle swarm optimization, ISA transactions 97 (2020) 415–430.

[33] X. Cao, H. Yan, Z. Huang, S. Ai, Y. Xu, R. Fu, X. Zou, A multi-objective particle swarm optimization for trajectory planning of fruit picking manipulator, Agronomy 11 (11) (2021) 2286.

[34] L. A. Márquez-Vega, M. Aguilera-Ruiz, L. M. Torres-Treviño, Multi-objective optimization of a quadrotor flock performing target zone search, Swarm and Evolutionary Computation 60 (2021) 100733.

[35] Z. A. Ali, Z. Han, R. J. Masood, Collective motion and self-organization of a swarm of uavs: A cluster-based architecture, Sensors 21 (11) (2021) 3820.

[36] A. Karimi, H. Nobahari, P. Siarry, Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions, Computational Optimization and Applications 45 (3) (2010) 639–661.

[37] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE transactions on Evolutionary Computation 6 (1) (2002) 58–73.

[38] Y. Shi, R. C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 1945–1950.

[39] G. Lin, Z. Han, C. Huepe, Order–disorder transitions in a minimal model of active elasticity, New Journal of Physics 23 (2) (2021) 023019.

[40] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, B. Lennox, Mona: an affordable open-source mobile robot for education and research, Journal of Intelligent & Robotic Systems 94 (3) (2019) 761–775.

[41] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, M. Mejail, A practical multirobot localization system, Journal of Intelligent & Robotic Systems 76 (2014) 539–562.

[42] S. Na, Y. Qiu, A. E. Turgut, J. Ulrich, T. Krajník, S. Yue, B. Lennox, F. Arvin, Bio-inspired artificial pheromone system for swarm robotics applications, Adaptive Behavior 29 (4) (2021) 395–415.

[43] Z. Ban, C. West, B. Lennox, F. Arvin, Self-organised flocking with simulated homogeneous robotic swarm, in: International Conference on Collaborative Computing: Networking, Applications and Worksharing, Springer, 2020, pp. 3–17.

[44] N. Eliot, D. Kendall, M. Brockway, A new metric for the analysis of swarms using potential fields, IEEE Access 6 (2018) 63258–63267.