Self-organised Flocking with Simulated Homogeneous Robotic Swarm

Zhe Ban[1(✉)], Craig West[2], Barry Lennox[1], and Farshad Arvin[1(✉)]

[1] Swarm and Computational Intelligence Lab (SwaCIL), Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK
{zhe.ban,farshad.arvin}@manchester.ac.uk

[2] Bristol Robotic Lab, University of West England, Bristol BS16 1QY, UK

Abstract. Flocking is a common behaviour observed in social animals such as birds and insects, which has received considerable attention in swarm robotics research studies. In this paper, a homogeneous self-organised flocking mechanism was implemented using simulated robots to verify a collective model. We identified and proposed solutions to the current gap between the theoretical model and the implementation with real-world robots. Quantitative experiments were designed with different factors which are swarm population size, desired distance between robots and the common goal force. To evaluate the group performance of the swarm, the average distance within the flock was chosen to show the coherency of the swarm, followed by statistical analysis to investigate the correlation between these factors. The results of the statistical analysis showed that compared with other factors, population size had a significant impact on the swarm flocking performance. This provides guidance on the application with real robots in terms of factors and strategic design.

Keywords: Swarm robotics · Flocking · Self-organised · Collective behaviour

1 Introduction

In nature, there are various collective motions commonly found in living organisms and social animals, such as shoals of fish [8], flocks of birds [3] and swarms of wildebeest [26]. Inspired by these collective motions, swarm robotics [20] was proposed as a research topic which provides collective strategies for a large number of simple robots to achieve collective behaviour. This collective behaviour potentially provides promising solutions to some problems in real life, such as, balancing the exploitation of renewable resources [15], fault detection [24], exploration in extreme environments [10] and coordination control of multiple autonomous cars [9]. To achieve these collective behaviours, a large and growing body of literature has investigated to model the swarm systems and to design relevant cooperation means. Considerable works have been undertaken from various angles for

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 350, pp. 3–17, 2021.
https://doi.org/10.1007/978-3-030-67540-0_1

different scenarios, such as distribution [6] and energy consumption [28]. Several coordination tasks have established, such as flocking [27], exploration [4], aggregation [1], foraging [23] and transportation [21]. Flocking is one the most important scenarios which has many real-world applications, e.g. in precision agriculture [5].

Cooperation strategies have a pivotal role in achieving flocking behaviour, hence, a number of strategies have been developed based on various disciplines to present collective motion and group behaviour. For example, Jia et al. [11] used a dominance matrix to compare between heterogeneous and homogeneous systems to propose a flocking framework with particles in different levels based on their contributions. In another study [7], disk graph and Delaunay graph methods were used to present connectivity with various distances. Also, mean-field game model was presented by partial differential equations to describe the system dynamics based on state and distributions [6]. The study by Thrun et al. proposed a cluster analysis that used the projection method based on the topographic map [25]. These strategies can be divided, on the basis of its framework, into two main categories: homogeneous and heterogeneous [11]. A heterogeneous group of swarm robots contains various types of robots with different roles and responsibilities, while in a homogeneous swarm, every individual follows the same strategy to achieve a common task, hence there are no behavioural or physical differences between the individuals in a swarm.

Developments of the strategies for flocking behaviour in a swarm system have led to a growing trend towards the real-world application of multi-robotic systems. Due to the limitations of real robots' hardware in practical scenarios, and based on the swarm robotics criteria defined by Şahin [19], each robot in a swarm system interacts with its direct neighbours within a specific range, *sensing radius*, to make decisions only based on its neighbours' stages. Direct communication in a swarm without having an extra observer is one of the challenges of implementing swarm scenarios using real mobile robots [13]. Some research studies [1,17] rely on the acquisition of the location of each robot from an extra observer, whereas some [2,11] regard each robot as an abstract particle without considering physical structures e.g. weight, size, motor speed and sensor range. Such approaches, however, can not realistically address the situation in which the group operation is influenced by the physical and hardware design constraints.

In one of the previous research studies [2], it has been theoretically demonstrated that Active Elastic Sheet (AES) is a self-propelled mechanism where swarm particles can successfully achieve collective motions. Due to the simplicity and robustness property of AES, in this paper, we chose this mechanism to demonstrate flocking behaviour of a homogeneous swarm. To combine hardware and collective control algorithms, the motion model was applied to a swarm of simulated robots by carefully considering the hardware limitations of the real robots. Hence, in this work, local communication relays on sensor values to make sure that the robots keep the desired distance from its neighbours. Each robot only detects the distance to its neighbour without acquiring the neighbour's identification, therefore, each robot is able to make decisions without an extra observer, in the whole process. Since the model of each individual robot includes all the physical properties which are carefully implemented by the control algorithm, the study here can be considered the first step from an abstract model to the progression with real robots. Simulated experiments were performed to analyse the group performance of the swarm flocking with the AES model. Followed by these qualitative experiments, the group behaviours were evaluated using a specified metric, i.e. the average distance between the robots in the swarm, and the results were statistically analysed to identify the effects of the chosen factors including time, population size and external (common) force. This information will potentially help follow-up studies to address several cautions of implementation using real swarm robots in real-world applications.

The rest of this paper was organised as follow. In Sect. 2, we introduced the collection formation and flocking mechanism. Following that, in Sect. 3, we explained the experimental setup and robotic platform. In Sect. 4, we discussed the experimental results and analysed effects of different parameters in collective swarm performance. Finally, in Sect. 5, we drew conclusions and discussed the future research direction in which the swarm robots might be involved.

2 Flocking Mechanism

The AES model [2] was originally developed and investigated using particles without hardware structures proposed. In this study, we utilised this model considering the hardware structure of physical robots. Constraints for each robot on position, $\dot{\boldsymbol{p}}_i$, and rotation, $\dot{\phi}_i$, are shown in Eqs. (1) and (2). The movement of each robot is controlled by two different forces: i) the goal force, $\boldsymbol{F_g}$, and ii) the collective force, $\boldsymbol{F_i}$. The goal force aims to steer the entire group moving towards a desired direction, while the collective force is used to keep robots within an expected distance to avoid the collision.

$$\dot{\boldsymbol{p}}_i = [\boldsymbol{F_g} + \alpha(\boldsymbol{F_i} + D_r\hat{\xi}_r) \cdot \hat{n}_i] \cdot \hat{n}_i, \tag{1}$$

$$\dot{\phi}_i = [\boldsymbol{F_g} + \beta(\boldsymbol{F_i} + D_r\hat{\xi}_r)] \cdot \hat{n}_i{}^{\perp}, \tag{2}$$

$$\boldsymbol{F_{gd}} = \gamma_d \boldsymbol{d}, \tag{3}$$

$$\boldsymbol{F_{gp}} = \gamma_p \hat{v}_i, \tag{4}$$

where coefficients α and β are related to linear speed and rotation of the collective movement. $\hat{n}_i$, $\hat{n}_i{}^{\perp}$ are unit vectors, where $\hat{n}_i$ has the same direction as the heading of the robot, while $\hat{n}_i{}^{\perp}$ is perpendicular to the heading direction. D_r is the noise value in the process of detecting distances between robots. $\hat{\xi}_r$ is a unit vector with a random direction, so that noise is applied in a arbitrary direction. ϕ_i is the angle which the robot i is expected to rotate. The clockwise direction is defined as positive and counterclockwise is negative for ϕ_i. γ_d and γ_p are the corresponding weight coefficients of goal forces. $\hat{v}_i$ is related to the desired group speed along the self-propulsive direction, which is proportional to the goal force, $\boldsymbol{F_{gp}}$. $\boldsymbol{F_{gd}}$ relies on the distance between the robot location and goal $\boldsymbol{d}$. These two sub-forces are shown as Eq. (3) and Eq. (4). The goal force, $\boldsymbol{F_g}$, consists of $\boldsymbol{F_{gp}}$ and $\boldsymbol{F_{gd}}$.

Each individual robot gets information about surrounding robots using their n sensors. The summation of sensors' values is presented as a collective force, $\boldsymbol{F_i}$, shown in Eq. (5):

$$\boldsymbol{F_i} = \sum_{j=1}^{n} -\frac{k\boldsymbol{r}_{ij}}{l_{ij}||\boldsymbol{r}_{ij}||}(||\boldsymbol{r}_{ij}|| - l_{ij}) \; , \tag{5}$$

where $\boldsymbol{r}_{ij}$ is the vector from the centre of the robot i to its neighbour j. Therefore, $||\boldsymbol{r}_{ij}||$ is the distance between robot i and j. l_{ij} is the desired distance between the two robots. The difference between the absolute value of $\boldsymbol{r}_{ij}$ and l_{ij} is the error of collective motion. $\frac{k}{l_{ij}}$ is a parameter which acts like a spring constant, involving the amount of force that robots generate according to the collective distances.

In terms of implementation of to the simulated robots, the process of flocking scenario is shown in Fig. 1. To begin this flocking scenario, each robot has an individual controller which is implemented by its own microcontroller. The robot uses six sensors to gain the surrounding information. These sensor readings are $||\boldsymbol{r}_{ij}||$ in Eq. (5). To steer the group of robots to a goal point or direction, we applied the desired distance, $\boldsymbol{d}$, for each robot. $\boldsymbol{F_g}$ and $\boldsymbol{F_i}$ have the same order of magnitude. After obtaining the force information, the controllers start to calculate total forces and correspondingly change the robot kinematics. The transformation from force to robot kinematics will be described in the Sect. 3.2. Once the motion property of each robot had been decided, the collective motion commences. Finally, the swarm collectively move to the goal position.

3 Experiments

3.1 General Foundation

Webots [14] is an open-source simulation software developed at the Swiss Federal Institute of Technology. With a 3D interface, the simulator provides numerous robotic modules and various objects, hence it is convenient to design swarm robotic systems. We used a miniature mobile robot *e-puck* [16] which is a popular swarm robotic platform and it has been utilised in many swarm robotics studies. The e-puck robot is equipped with two differential driven wheels for its actuation, and eight infra-red (IR) sensors for proximity measurements which are mainly used in our experiments for decision making. Figure 2 shows an e-puck model and its top view in Webots.

From the top view, each e-puck has eight horizontally symmetrical sensors, but not vertically symmetrical, as there are two more sensors in the front part.

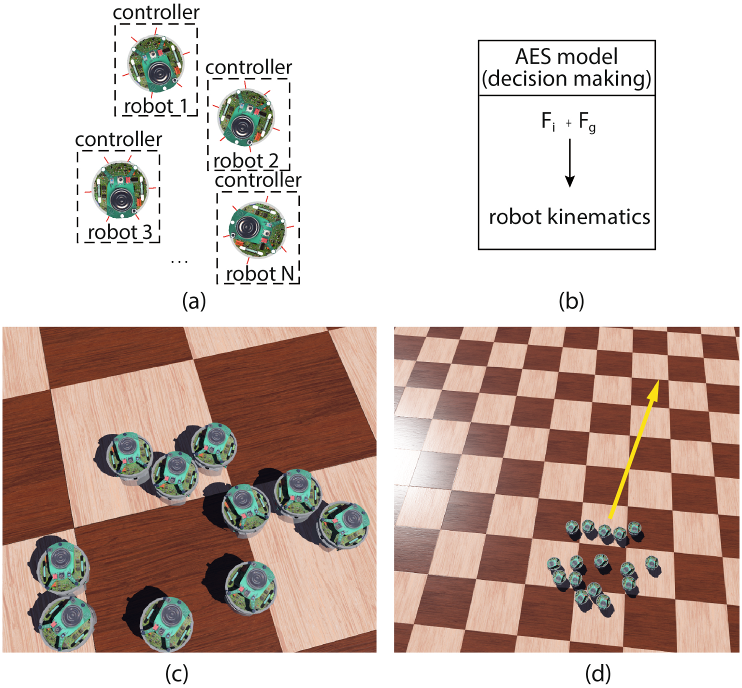


Fig. 1. (a) First step: each robot detects neighbours using sensors, which is also the data collection for calculation of collective force, $\boldsymbol{F_i}$. Every robot has a controller to collect and calculate their own data. (b) Second step: In every individual controller, total force is calculated based on $\boldsymbol{F_g}$ and $\boldsymbol{F_i}$ using the AES model. This is also a step for each robot deciding how to change their kinematics. (c) Third step: controllers change the speed of actuators to change the robots' motion. (d) Fourth step: flocking motion achieved. The picture is a screenshot of an experiment, where the swarm collectively moved to an expected direction along the yellow arrow. (Color figure online)

Since the forces in the AES model depend on the sensor values in the experiments, the distribution of sensors has a significant impact on decision-making for each robot. Asymmetrical distribution of sensors gives rise to unbalanced forces. To balance the distribution of force, six sensors are chosen which are marked with red colour in Fig. 2 (a), correspondingly $n = 6$ in Eq. (5). Due to the fixed hardware design, the distribution of sensors is still slightly asymmetrical. According to the provided documentation[1], the infra-red sensors in each e-puck have a deviation of noise which obeys a Gaussian random distribution.

In each robot, the maximum rotation speed of the motor is 6.28 rad/s. With 20.5 mm wheel radius, the maximum speed of an e-puck is 0.25 m/s. Thanks to the fully integrated cross-compilation in Webots for the e-puck, less modification is needed for the controller from simulation to the real robots implementation.

[1] http://www.cyberbotics.com.

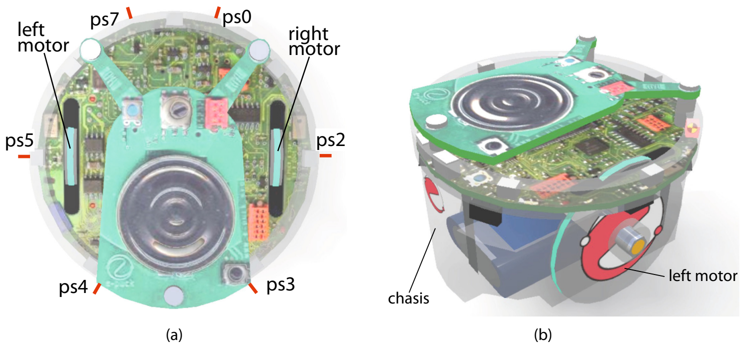


Fig. 2. (a) Top view of an e-puck robot with upward heading direction. The red lines extended from proximity sensors represent the perception ranges. Given the zero angle as the positive x direction and moving anticlockwise is taken as positive, the orientation of sensor $\{0, 2, 3, 4, 5, 7\}$ are $\{73°, 0°, 300°, 240°, 180°, 107°\}$. The intervals between them are not all the same, which will lead to the unbalanced weight from different directions. (b) 3D model of e-puck in Webots. The e-puck is differentially driven by two motors. (Color figure online)

Compared with the particle based simulations, this simulation is closer to the real-world scenarios and the controllers can be easily transferred to the real robots.

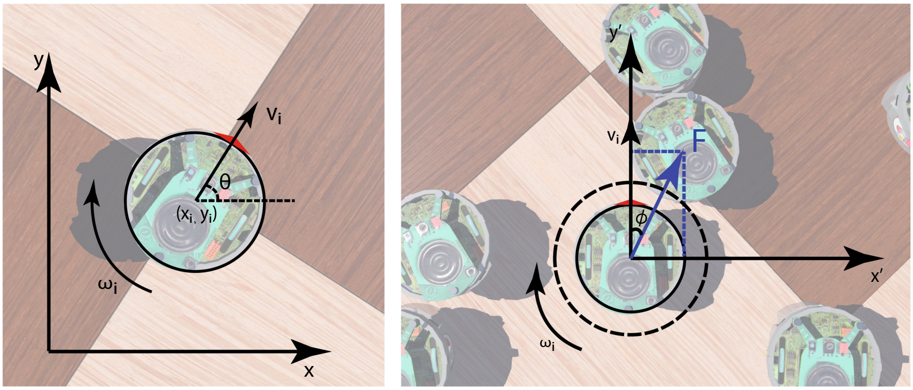
3.2 Individual Robot Test

Precise tracking performance of an individual robot is the foundation of accurate group behaviour. Since each robot basically makes decisions based on multiple forces in the AES model, we designed an individual robot test to improve the motion of a single robot before group experiments. The robot needed to make a trade-off between agility and accuracy under a force with an arbitrary direction. In this test, the relevant parameters γ_p needed to be tuned, so that the robot makes reliable decisions.

In Fig. 3(a), the pose of the robot i is described as $(x_i,\ y_i,\ \theta_i)$. In Fig. 3(b), x_i and y_i represent the expected movement of the robot which is related to the projection of $\boldsymbol{p}_i$ onto the x and y-axes. Forward velocity, $\hat{v}_i$, and angular velocity, w_i, are two variables to describe the kinematics of the robot. Equation (6) presents the transformation from the kinematics to the pose of the robot:

$$\dot{x}_i = ||\hat{v}_i|| \cos\theta_i, \quad \dot{y}_i = ||\hat{v}_i|| \sin\theta_i, \quad \dot{\theta}_i = w_i. \tag{6}$$

Since e-puck is a typical differential wheeled robot, its motion depends on the speeds of both left and right wheels. The forward velocity of the e-puck is given by the average speed of both wheels, while the rotational velocity is related to the differences between the speed of two wheels. Therefore, the position and orientation of an e-puck can be presented by:



(a) (b)

Fig. 3. (a) An e-puck and its state: the red arrow at the top of the robot indicates the heading of e-puck. v_i and w_i denote the forward velocity and angular velocity of e-puck i. The forward velocity is in the same direction as the heading of e-puck. Angular velocity is positive when e-puck rotates clockwise. (b) The coordinate of an e-puck: the x-axis is along the rotation axis of the wheels. The heading of e-puck is in the same direction as the y-axis. The black circle with the dashed line denotes the range of sensor measurement. The blue arrow denotes the force acting upon the e-puck, which is projected onto x and y-axis for calculation. All the forces are calculated according to this coordinate. In this figure, the force is from the nearby robots. By comparing this force and expected one, the robot was changing its kinematics. In the background, the swarm of mobile robots wandered to the goal as a coherent group in an open space. (Color figure online)

$$\dot{x_i} = r(\frac{w_r + w_l}{2})\cos\theta_i, \quad \dot{y_i} = r(\frac{w_r + w_l}{2})\sin\theta_i, \quad \dot{\theta_i} = r(\frac{w_r - w_l}{l}), \tag{7}$$

where w_r and w_l denote rotational speeds of right and left wheels respectively, l is the distance between the wheels, and r is the radius of the wheels.

Then, the transformation between pose goal and the angular velocity of wheels can be derived by combining Eq. (6, 7):

$$\begin{bmatrix} ||\hat{v}_i|| \\ w_i \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix}. \tag{8}$$

According to the specification of the e-puck, $r = 20.5$ mm and $l = 52$ mm, the rotational speeds of the left and right wheels follow:

$$w_l = 487.8049\ ||\hat{v}_i|| - 1.2683 w_i, \tag{9}$$

$$w_r = 487.8049\ ||\hat{v}_i|| + 1.2683 w_i. \tag{10}$$

A coordinate of a robot designed for the change of kinematics is shown in Fig. 3(b). The origin is the centre of the e-puck and y-axis is along with the robot's heading. We projected a force onto the x-y plane. The projection has been included here for two reasons: i) the forces are vectors in the AES model and projection can transfer the information to desired pose $(x_i, y_i,\ \theta_i)$ which will be mentioned below and ii) it is simple to be implemented in C/C++ programming language.

According to the AES model, $\hat{v}_i$ and w_i are related to $\boldsymbol{F_i}$ and $\boldsymbol{F_g}$, which can be calculated as:

$$||\hat{v}_i|| = m\ ||\boldsymbol{F_i} + \boldsymbol{F_g}|| \quad , \quad w_i = n\angle(\boldsymbol{F_i} + \boldsymbol{F_g})\ , \tag{11}$$

where parameter m and n were tuned empirically to make sure each e-puck rapidly adjust its speed and heading, which is the fundamental of group behaviour.

3.3 Swarm Robots Test

In this section, we aimed to apply the AES model for the swarm. In order to test the group's behaviour with different kinds of initial situations, the start orientation of each agent was randomly set $\theta \in [-\pi, +\pi]$. The distances between nearby robots were less than half of the sensor range to make sure robots were able to detect their neighbours.

According to the AES model, there are two forces that mainly affect the collective motion. One is the collective force, $\boldsymbol{F_i}$, and the other is the goal force, $\boldsymbol{F_g}$. Collective force depends on the relative positions of the neighbours, which can be calculated by summing the errors between sensor values, r_{ij}, and desired distances, l_{ij}. For each robot, six forces responding to six sensor values are added up to find the resultant force, $\boldsymbol{F_i}$, and each sensor value is regarded as a vector. The direction of the vectors are from the centre of the e-puck to the corresponding sensors, which are along the red lines in Fig. 2(a). The blue arrow in Fig. 3(b) is an example which illustrates the total collective force in the group test. In terms of the goal force, the magnitude of $\boldsymbol{F_g}$ is a constant value, which pulls the swarm toward a pre-set direction.

Dot products are used to calculate the scalar projection of the forces onto a horizontal unit vector $\hat{n}_i$ and a vertical unit vector $\hat{n}_i{}^{\perp}$ in Eqs. (1) and (2). In our work, the angles between $\hat{n}_i$ and forces depend on the distribution of sensors in each e-puck coordinate system. There are seven projections calculated for each e-puck, including six forces from sensors and a goal force, $\boldsymbol{F_g}$. The component of the total force acting in the horizontal direction was calculated by summation of the component of all forces in $\hat{n}_i$ direction as presented in Eq. (5). The component of the total force in the vertical direction was calculated in the same way using $\hat{n}_i{}^{\perp}$. Prior to applying the goal force, the coordinate transformation was adopted because $\boldsymbol{F_g}$ are in the global coordinate frame, while the set of $\boldsymbol{F_i}$ controller on the e-puck is in robot coordinate frame.

Collective and goal forces have weight parameters α, β and γ_p, γ_d, w_l, respectively. These parameters influence the forces that are applied to the robot, for example, an increase in γ_p leads to a bigger force to pull the swarm toward a direction and vice versa. As a homogeneous robotic model, each robot was deployed the same controller to achieve decentralised collective motion. In this study, each experiment contained 336 positions for each e-puck. With the established individual behaviour and the calculation of forces, the flocking behaviour has been achieved. The source codes for all implementations are available on GitHub[2].

3.4 Metrics

Flocking behaviour is a simultaneous motion where a group of robots move toward a target direction. The likelihood of individuals remaining in the group depends on the coherency of the swarm. Here, we focus on the cohesiveness of the swarm. To evaluate the swarm coherency, the average distance between the swarm members is calculated as a metric in this study, which is a common method has been used in many research studies, e.g. in [17,18]. The average distance in this paper is the mean value of the distances between robots, which can be calculated as:

$$d_s = \frac{2\sum_{i=1}^{N-1}\sum_{j=1}^{N}||\boldsymbol{r}_{ij}||}{N(N-1)}\ , \tag{12}$$

where N is the number of robots in the group.

Analytical experiments were conducted to show the impact of several factors including forces, population size and desired distance on the d_s. Table 1 gives a summary of the details about factors, including different number population sizes $N \in \{4, 6, 9, 12\}$ robots, goal forces $\gamma_p \in \{2000, 3000\}$ and desired distance of $l_{ij} \in \{120, 150\}$. The desired distance l_{ij} here is dimensionless, because the value is related to sensor values. 16 sets of experiments were run and each set of experiments were repeated 10 times. To facilitate calculation of the metrics, d_s, it is important to accurately ascertain each robot's location at each sampling time.

Table 1. Sets of experiments for analysis of factors in each population size.

γ_p , l_{ij}	**4**	**6**	**9**	**12**
$\gamma_p = 2000$, $l_{ij} = 120$	Set1	Set2	Set3	Set4
$\gamma_p = 2000$, $l_{ij} = 150$	Set5	Set6	Set7	Set8
$\gamma_p = 3000$, $l_{ij} = 120$	Set9	Set10	Set11	Set12
$\gamma_p = 3000$, $l_{ij} = 150$	Set13	Set14	Set15	Set16

[2] https://github.com/swacil/Flocking.

4 Result and Discussion

In order to identify the impacts of factors N, l_{ij} and γ_p on the group performance, a series of tests were selected from the above experiments, in which only the object factor is an independent variable and other factors are fixed. Box-plots were used to show the distributions of distances with different populations N, sensor sensitivity l_{ij} and group force γ_p.

4.1 Effects of Time

Each experiment took 22 s and contained 336 sample points for each e-puck. In an attempt to simplify data analysis and comparison, every 28 data was averaged to represent the average distances, d_s, in every two seconds. As shown in Fig. 4(a), the median of the d_s held steadily around the initial value for the first 14 s that indicates the swarm was able to maintain the coherency as the initialisation within this period of time. However, distribution of the obtained results (size of boxes) increases over time hence it is evident that the maximum d_s increases, namely d_s had a greater chance to reach the maximum value of a test at the end of the experiment. During the experiments, the swarm spread apart as time goes on.

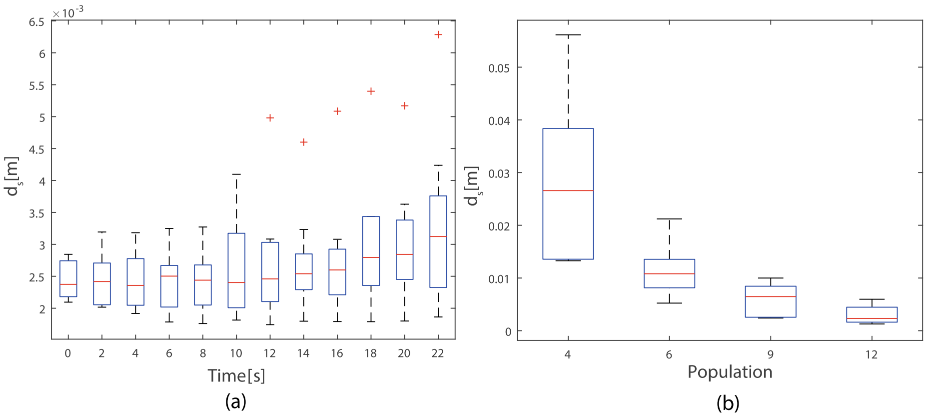


Fig. 4. (a) Box plots of average distance during experiments with $F_g = 3000$, $l_{ij} = 120$ using 12 robots within 22 s. (b) Box plots of average distance during experiments with $F_g = 2000$, $l_{ij} = 120$ using swarms with different population sizes.

4.2 Effects of Population

Figure 4(b) shows the swarm coherency, d_s, with $\gamma_p = 2000$, desired distance of $l_{ij} = 120$ and varying swarm size of $N \in \{4, 6, 9, 12\}$ robots. Considering the results of the experiments with the same goal force and the desired distance, the minimum and maximum d_s see declines as the population increase. It can be clearly observed that the median d_s decreases as population increases. All in all, the swarm with a larger population yields better group performance when other factors are the same.

4.3 Group Force

To compare group performance under different group forces, the data of the first and third row in Table 1 was chosen. The d_s with $l_{ij} = 120$ within 19 s are shown in Fig. 5. Comparing the d_s with $\gamma_p = \{2000, 3000\}$, there is a dramatic rise of the median and maximum d_s in all experiments, while the minimum d_s saw a decline apart from the swarm with 6 robots. Overall, the bigger force triggers larger d_s, resulting in worse coherency regardless of population size.

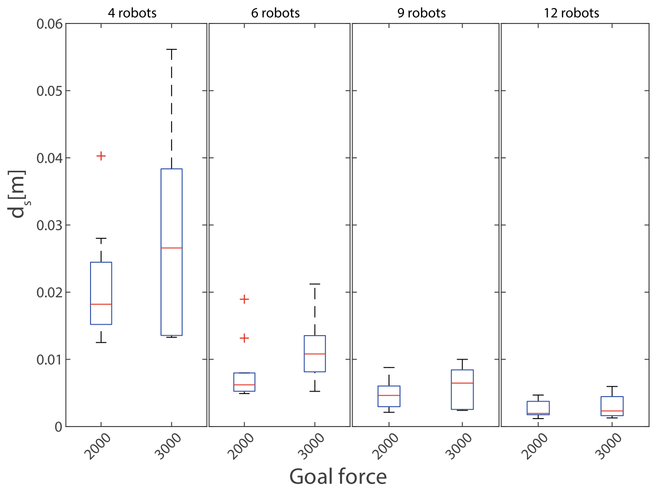


Fig. 5. Box-plots of average distance during experiments with $l_{ij} = 120$ and $N \in \{4, 6, 9, 12\}$ robots using different goal forces.

4.4 Desired Distance

Desired distance, l_{ij}, is the target distances between the robots, which is directly related to a specific sensor value in the simulation. We varied l_{ij} and kept $\gamma_p = 2000$ in the first and second row of experiments in Table 1. Figure 6 illustrates that a bigger desired distance leads to larger d_s for small population size. In contrast, bigger desired distance contributes to smaller d_s when the swarm has a larger population sizes.

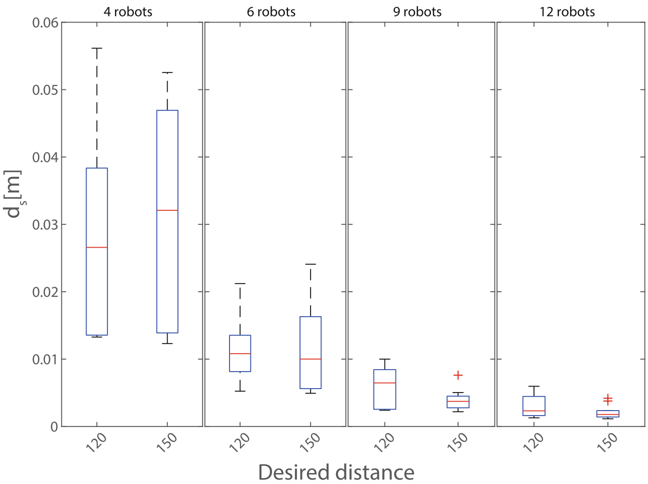


Fig. 6. Box plots of average distance during experiments with $F_g = 2000$ and $N \in \{4, 6, 9, 12\}$ robots using different desired distances.

4.5 Statistical Analysis

To compare the influence of the aforementioned factors, statistical analysis was made using the data in Table 1. According to the result of Kruskall-Wallis test, the data were normally distributed. As a common data analysis method, Analysis of Variance (ANOVA) test [22] was adopted to assess if the factors have a statistically significant effect on group behaviour. In the F-test, the three parameters: force, desired distance and population size are the factors and d_s is the result to represent the group behaviour. According to the properties of F-distributions, traditionally, $p = 0.05$ is chosen as a significance level [12]. When the p-value is less than the significance level, the null hypothesis is rejected and the corresponding factors have significant impacts on the result.

Results of the ANOVA are shown in Table 2. The number of robots plays the most significant role in coherency because p-value of the population size is far less than 0.5 ($p_p \approx 0.000$). Increasing the population size contributes to better group performance significantly. Though desired distance has an impact on the coherency of the swarm, the influence is less than the population size and force. Force has the least impact on coherency with $p_p \approx 0.691$. Compared with a swarm with fewer robots, large-sized swarms tend to have more coherent performance even under different goal forces and desired distances.

4.6 Discussion

This study showed that, an increase in population size of a swarm improve the collective performance. This is one of the main criteria of swarm robotics [19], which has been also reported in many studies [1,10,17,18]. This shows that the

Table 2. Results of analysis of variance (ANOVA).

Factors	p-value	F-value
Force (F_g)	0.912	0.015
Desired distance (l_{ij})	0.691	0.381
Population (N)	0.000 (<0.001)	33.039

increased inter-robot interactions due to the higher population size resulted in improvement of collective behaviour of the swarm.

Also, it is interesting to note that the robots' ID is not necessary for all experiments during the decision-making process. Since the model here is decentralised and every controller of the robot is independent, each individual enables to make decisions with stochastic neighbours by sensor detection.

In some of the previous works [9,17], collective motion of a swarm relies on an extra observer, such as a camera or simulation platform. A portion of robots make decisions according to the specific robots' information which provides by the extra observer. Compared with these works, we provide a solution with less requirement of equipment and better tolerance for the change of neighbours when the accuracy of the group performance is less regulated.

The decision-making fully relies on sensors in this study, hence the sensitivity of sensor affects group performance. One source of weakness in this study which could have affected the group behaviour is sensor distribution. As mentioned in Sect. 3.1, e-puck has asymmetry sensor distribution which led to the unbalance of sensor values between the front and back. Also, this study did not consider an environment with obstacles and only focused on the flocking behaviour in an open space. Besides, the scope of this study was limited in terms of evaluation of group performance. Average distance is the only metric to describe the density of the swarm. The speed of group motion and group size were the paucities of the study.

5 Conclusion

This work was undertaken to implement a collective motion model using a realistic simulated robot swarm. We evaluated the influence of several factors on the swarm performance by quantitative experiments. The simulated robots were able to achieve flocking behaviour as a single, cohesive group. The results of the study demonstrated that the population size plays a significant role in flocking behaviour. The insights gained from this study may be of assistance to applying an abstract model to the real-world robotic applications. A natural progression of this work is to transfer the simulation to a real robotic scenario. A further study will investigate the application of this flocking mechanism in the inspection of farms for precision agriculture.

Acknowledgments. This paper was partially supported by the UK EPSRC RAIN (EP/R026084/1) and RNE (EP/P01366X/1) projects.

References

1. Arvin, F., Turgut, A.E., Krajník, T., Yue, S.: Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm. Adaptive Behav. **24**(2), 102–118 (2016)
2. Ferrante, E., Turgut, A.E., Dorigo, M., Huepe, C.: Collective motion dynamics of active solids and active crystals. New J. Phys. **15**(9), 095011 (2013)
3. Flack, A., Nagy, M., Fiedler, W., Couzin, I.D., Wikelski, M.: From local collective behavior to global migratory patterns in white storks. Science **360**(6391), 911–914 (2018)
4. Gifford, C.M., et al.: A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping. Robot. Auton. Syst. **58**(2), 186–202 (2010)
5. Grieve, B.D., et al.: The challenges posed by global broadacre crops in delivering smart agri-robotic solutions: a fundamental rethink is required. Global Food Secur. **23**, 116–124 (2019)
6. Grover, P., Bakshi, K., Theodorou, E.A.: A mean-field game model for homogeneous flocking. Chaos: Interdisc. J. Nonlinear Sci. **28**(6), 061103 (2018)
7. He, C., Feng, Z., Ren, Z.: A flocking algorithm for multi-agent systems with connectivity preservation under hybrid metric-topological interactions. PloS One **13**(2), e0192987 (2018)
8. Hein, A.M., Gil, M.A., Twomey, C.R., Couzin, I.D., Levin, S.A.: Conserved behavioral circuits govern high-speed decision-making in wild fish shoals. Proc. Natl. Acad. Sci. **115**(48), 12224–12228 (2018)
9. Hu, J., Bhowmick, P., Arvin, F., Lanzon, A., Lennox, B.: Cooperative control of heterogeneous connected vehicle platoons: an adaptive leader-following approach. IEEE Robot. Autom. Lett. **5**(2), 977–984 (2020)
10. Huang, X., Arvin, F., West, C., Watson, S., Lennox, B.: Exploration in extreme environments with swarm robotic system. In: IEEE International Conference on Mechatronics (ICM), vol. 1, pp. 193–198 (2019)
11. Jia, Y., Vicsek, T.: Modelling hierarchical flocking. arXiv preprint arXiv:1904.09584 (2019)
12. Laubscher, N.F., et al.: Normalizing the noncentral t and f distributions. Ann. Math. Stat. **31**(4), 1105–1112 (1960)
13. Liu, Z., West, C., Lennox, B., Arvin, F.: Local bearing estimation for a swarmof low-cost miniature robots. Sensors **20**(11) (2020)
14. Michel, O.: Cyberbotics ltd. webots™: professional mobile robot simulation. Int. J. Adv. Robot. Syst. **1**(1), 5 (2004)
15. Miletitch, R., Dorigo, M., Trianni, V.: Balancing exploitation of renewable resources by a robot swarm. Swarm Intell. **12**(4), 307–326 (2018)
16. Mondada, F., et al.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, vol. 1, pp. 59–65 (2009)
17. Na, S., et al.: Bio-inspired artificial pheromone system for swarm robotics applications. Adaptive Behav. 1–21 (2020)

18. Na, S., Raoufi, M., Turgut, A.E., Krajník, T., Arvin, F.: Extended artificial pheromone system for swarm robotic applications. In: The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE), pp. 608–615. MIT Press (2019)
19. Şahin, E.: Swarm robotics: from sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics. SR 2004. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30552-1_2
20. Schranz, M., Caro, G.A.D., Schmickl, T., Elmenreich, W., Arvin, F.,Şekercioğlu, A.: Swarm intelligence and cyber-physical systems: concepts, challenges and future trends. Swarm Evol. Comput. (2020)
21. Shao, J., Wang, L., Yu, J.: Development of an artificial fish-like robot and its application in cooperative transportation. Control Eng. Practice **16**(5), 569–584 (2008)
22. Tabachnick, B.G., Fidell, L.S.: Experimental Designs Using ANOVA. Thomson/Brooks/Cole Belmont, CA (2007)
23. Tarapore, D., Floreano, D., Keller, L.: Influence of the level of polyandry and genetic architecture on division of labour. In: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, pp. 358–364. No. CONF, MIT Press (2006)
24. Tarapore, D., Timmis, J., Christensen, A.L.: Fault detection in a swarm of physical robots based on behavioral outlier detection. IEEE Trans. Robot. **35**(6), 1516–1522 (2019)
25. Thrun, M.C.: Projection-Based Clustering Through Self-organization and Swarm Intelligence: Combining Cluster Analysis with the Visualization of High-dimensional Data. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-658-20540-9
26. Torney, C.J., Hopcraft, J.G.C., Morrison, T.A., Couzin, I.D., Levin, S.A.: From single steps to mass migration: the problem of scale in the movement ecology of the serengeti wildebeest. Philos. Trans. Roy. Soc. B: Biol. Sci. **373**(1746), 20170012 (2018)
27. Turgut, A.E., Çelikkanat, H., Gökçe, F., Şahin, E.: Self-organized flocking in mobile robot swarms. Swarm Intell. **2**(2–4), 97–120 (2008)
28. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. **75**(6), 1226 (1995)