Self-Organised Collision-Free Flocking Mechanism in Heterogeneous Robot Swarms

Zhe Ban[1] · Junyan Hu[1] 🔾 · Barry Lennox[1] · Farshad Arvin[1]

Abstract

Flocking is a social animals' common behaviour observed in nature. It has a great potential for real-world applications such as exploration in agri-robotics using low-cost robotic solutions. In this paper, an extended model of a self-organised flocking mechanism using heterogeneous swarm system is proposed. The proposed model for swarm robotic systems is a combination of a collective motion mechanism with obstacle avoidance functions, which ensures a collision-free flocking trajectory for the followers. An optimal control model for the leader is also developed to steer the swarm to a desired goal location. Compared to the conventional methods, by using the proposed model, the swarm network has less requirement for power and storage. The feasibility of the proposed self-organised flocking algorithm is validated by realistic robotic simulation software.

Keywords Swarm robotics · Flocking · Self-organised · Collective behaviour · Obstacle avoidance

1 Introduction

Flocking, a collective motion of individuals with a limited communication ability, is a social animals' common behaviour observed in nature [1]. There are many types of collective motions in nature mainly found in living organisms and social animals, such as shoals of fish [2], flocks of birds [3] and swarms of wildebeest [4]. In addition, Kaiser et al. [5] have demonstrated that microscopic colloidal particles show collective motion in presence of magnetic fields. Inspired by the collective behaviours, swarm robotics has been emerged as a research topic that provides collective strategies for a large number of simple robots in order to achieve fascinating collective behaviours [6, 7]. These collective behaviours potentially provide promising solutions to real-life problems, e.g. balancing the exploitation of renewable resources [8], autonomous swarm

shepherding [9], fault detection [10], exploration in extreme environments [11, 12] and coordination control of multiple autonomous vehicles [13].

To achieve the collective behaviours and to develop collective mechanisms, a large and growing body of literature has investigated modelling and designing the swarm systems. Considerable works have been undertaken from various angles such as distribution [14], communication [15], sensing [16] and coordination [17]. As a significant aspect of swarm systems, cooperation strategy particularly plays an important role in successfully achieving flocking behaviour. Hence, different strategies have been developed based on various disciplines to present collective motion as a connected swarm behaviour, for example, in the study conducted by He et al. [18], disk graph and Delaunay graph methods were used to present connectivity with various distances. In [14], mean-field game model was presented by partial differential equations to describe the swarm dynamics based on state and distributions. The performed study by Thrun et al. [19] proposed a cluster analysis that used the projection method based on the topographic map. These strategies can be divided, on the basis of its framework, into two main categories: homogeneous and heterogeneous [20]. 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 the same task, hence, there are

---

✉ Junyan Hu
  junyan.hu@manchester.ac.uk

  Farshad Arvin
  farshad.arvin@manchester.ac.uk

1 Swarm & Computational Intelligence Lab (SwaCIL),
   Department of Electrical and Electronic Engineering,
   The University of Manchester, Manchester, UK

no behavioural or physical differences between the individuals in a swarm [21]. A heterogeneous model is used in this research study to achieve centralised decision making for a swarm of robots.

Bio- and nature-inspired flocking mechanisms were investigated in many studies, e.g. Reynolds flocking model [22] is one of the fundamental swarm flocking presentations. In another study [23], a *leaderless* mechanism was proposed, in which robots estimated swarm's centre-of-mass using neighbouring robots detection within their sensing radius. Campo et al. [24] proposed a target estimation method based on a negotiation within the swarm during a flocking scenario. The study by Turgut et al. [25] set up a self-organised flocking using real mobile robots which were locally connected using wireless network. In addition, a collective motion with combination of aggregation and velocity alignment was proposed [26]. In [27], collective motion was investigated under two conditions that are i) agents follow their own polarity and ii) agents interact with each other. Moreover, a flocking mechanisms was proposed in [28] using magnitude-dependent motion control to steer a swarm. In a recent study, Na et al. [29] presented a bio-inspired collective migration of a swarm that were steered by movement of an external cue embodied in an artificial pheromone. In a seminal study in collective motion, Ferrante et al. [30] proposed a flocking method based on active elastic sheet (AES), which is a self-propelled mechanism where swarm particles can successfully achieve collective motions. The aforementioned studies are examples of collective motion mechanisms which were mainly inspired by nature.

On the other hand, theoretical studies based on multi-agent systems have been conducted for several decades. In [31], a consensus algorithm was developed for multi-agent systems under fixed and switching communication topologies. The synchronisation of the agents can be achieved via only local information. In [32], a robust consensus control scheme was designed for voltage and frequency restoration of islanded microgrids, where the effects of model uncertainty, parameter variation and unmodelled dynamics were considered. Time-varying delays were also considered in the leader-following consensus problems as shown in [33], where a condition to reach consensus and an estimation of the convergence rate were given. Based on the progress of the research in the area of multi-agent consensus control, various formation control techniques have been developed for swarm systems to handle more practical applications. In [34], formation tracking of networked mobile robots using bearing measurements was proposed. In another work [35], a two-layer formation-containment coordination scheme was proposed and an application to artificial satellites formation flying was shown. In [17], robust formation coordination of robot swarms was analyzed, where the nonlinear

dynamics and unknown disturbances were considered in the coordination protocol design. As shown in the aforementioned literature, multi-agent systems play an important role in coordinating robots to accomplish the desired goals especially in real-world applications.

Typically, nature-inspired swarm strategies are simple but offer less precise swarm behaviour. In contrast, multi-agent systems theory generally provide more accurate control for a swarm, but has a higher computational cost. A well-designed heterogeneous swarm system, such as leader-follower mechanism, is able to combine the advantages of nature-inspired swarm systems with multi-agent systems theory. Using a leader-follower mechanism, where a leader acts as a local central controller and followers act as a swarm system dealing with the target task e.g. weeding, will provide great opportunity for swarm systems in real-world applications. This means, a leader, an expensive robot equipped with various communication and sensing tools, plays a base transceiver stations role in a swarm, while followers are low-cost individuals that receive high level commands, e.g. steering the swarm, from the leader. However, one of the common challenges in the leader-follower mechanism is distributing tasks. Providing a model to allocate efficient tasks for leader and follows is crucial, which may largely decrease the overall mission completion time and energy costs and even further lower the requirement for hardware. Besides, how to improve the robustness of a swarm is also a challenge in swarm robotics [20]. An ideal swarm mechanism should be able to provide a solution to ensure the majority of robots achieve the goal even when a small amount of robot get lost.

In this paper, we proposed a leader-follower mechanism combining a low-cost heterogeneous swarm (for the flock) and a leader coordination control. The proposed leader-follower mechanism is tailored for the real-world applications of swarm robotics, to reduce the cost of implementation by deploying a small number of costly robots that are coordinating the entire swarm system. To combine hardware and collective control algorithms, the motion model was applied to a swarm of simulated robots by carefully considering the physical properties and hardware limitations of the real robots. Simulated experiments were performed to analyse the group performance of the swarm flocking with the leader-follower mechanism. Followed by these qualitative experiments, the group behaviours were evaluated using two metrics, i.e. the average distance between the robots and the swarm alignment in the swarm. The results were analysed to identify the effects of the chosen factors including time, population size, force and desired distance. This information will potentially help follow-up studies to address limitations of implementation using real swarm robots in real-world applications.

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

2 Flocking method

The proposed flocking method, including the swarm controller and leader's coordination control, is presented in this section. Apart from the leader, every robot follows the swarm mechanism.

2.1 Swarm mechanism

In the swarm mechanism, we designed a strategy to shape a swarm and build a network, so that the robots interact with each other. Meanwhile, we proposed a solution for obstacle avoidance which is a critical topic in application of swarm robotics in real-world scenarios. As a result, the swarm not only can perform the collective motion in an open space, but also is able to avoid obstacles and maintain the group after the obstacle avoidance task. First, we described the system as Eqs. 1 and 2 which show the description of each robot on position, $\dot{\vec{p}}_i$, and robot rotation, $\dot{\phi}_i$. The movement of each robot is controlled by two different forces: i) the repulsive force, $\vec{F}_o$, and ii) the collective force, $\vec{F}_i$, described as Eqs. 3 and 4.

$$\dot{\vec{p}}_i = [\vec{F}_o + \alpha(\vec{F}_i + D_r\hat{\xi}_r) \cdot \hat{n}_i] \cdot \hat{n}_i , \tag{1}$$

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

where coefficients α and β are related to linear speed and angular speed of the collective movement. $\hat{n}_i$ and $\hat{n}_i^{\perp}$ 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 an arbitrary direction. ϕ_i is the angle which the robot i is expected to rotate. A positive value of ϕ_i determines a clockwise rotation and a negative value is a counterclockwise rotation. $\vec{F}_o$ and $\vec{F}_i$ are described as follows

$$\vec{F}_o = \sum_{o=1}^{n_o} -\frac{k_o\vec{r}_o}{l_o\|\vec{r}_o\|}(\|\vec{r}_o\| - l_o) , \tag{3}$$

$$\vec{F}_i = \sum_{j=1}^{n_n} -\frac{k\vec{r}_{ij}}{l_{ij}\|\vec{r}_{ij}\|}(\|\vec{r}_{ij}\| - l_{ij}) . \tag{4}$$

$\vec{F}_o$ presents the force which obstacles exert to the robot and the repulsive force $\vec{F}_o$ is related to the differences between $\|\vec{r}_o\|$ and l_o. $\vec{r}_o$ is the vector from the centre of the robot to the obstacle o, so $\|\vec{r}_o\|$ is the distance between the robot and obstacle o. l_o is the threshold of avoiding action. n_o is the total number of obstacles. Obviously, the repulsive force is zero when the distance between robot and obstacle equals to the threshold. The force increases as the robot move close to the obstacle.

The repulsive force is only applied when the robot near an obstacle, which is shown in Algorithm 1. The repulsive force aims to avoid obstacles for each robot on their path, while the collective force is used to build a network among robots.

Algorithm 1 Obstacle avoidance.

Initialisation of N; k; l_{ij}; l_o
while *true* **do**
    for $o \leftarrow 1$ *to* n_o **do**
        if $\vec{r}_o > 0.4$ **then**
            $\vec{F}_o = 0$;
        else
            $\vec{F}_o$= the value of Eq. 3;
        end
    end
end

Similarly, the vectorial summation of the difference between neighbouring distance and desired distance is presented as a collective force, $\vec{F}_i$, shown in Eq. 4. n_n is the number of neighbours. $\vec{r}_{ij}$ is the neighbouring robot which is a vector from the centre of the robot i to its neighbour j. Therefore, $\|\vec{r}_{ij}\|$ is the distance between robot i and j. l_{ij} is the desired distance between two neighbouring robots. The difference between the absolute value of $\vec{r}_{ij}$ 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.

Remark 1 In some relevant works, obstacle avoidance was considered in the leader-following formation control design. For example, in [36], a leader robot travelled across an area with obstacle avoidance and a follower robot followed the exact same trajectory to avoid these obstacles. Bluetooth technology was used for communication between the two robots. In another study conducted by Lee et al. [37], three boundary layers were applied to achieve a local leader-follower method. Different layers were prioritised to keep the shape of the group of robots and avoid the collision according to the distance between robots and obstacles. Apart from the leader robot, each robot followed its front neighbour. In this case, the decision of each individual robot completely relies on one neighbour robot and the swarm

acts as a chain. Gain control and corrective control were adopted to set the trajectory for obstacle avoidance in [38], where Wu et al. designed a formation tracking task using one leader robot and two follower robots in the simulation. However, in the aforementioned works, one disconnection between two robots may result in the loss of the rest of the robots in the chain. Besides, as all followers require the information from the leader, the communication costs are high when a practical swarm has a large size and some robots are far away from the leader. Different from these works, in this paper, followers make decisions only based on their local neighbours within the sensing range. Therefore, if a small amount of robots lose connections, the majority of the swarm are still able to achieve the goal.

2.2 Leader motion

In this section, the motion of leader is analysed. It is assumed that the target position of the swarm can only be detected by the leader. In order to navigate all the followers to the goal, we designed the following tracking method for the leader based on the relative position information.

Consider that many robotic systems (e.g. mobile robots) can be transformed into double integrator dynamics via input-output feedback linearisation techniques [39, 40]. In this work, the dynamics of the leader robot can be described by the following equations

$$\dot{p}(t) = v(t), \tag{5}$$
$$\dot{v}(t) = u(t), \tag{6}$$

where $p(t)$ and $v(t)$ represent the position and velocity of the robot respectively. $u(t)$ is the control input signal to be designed. This dynamic model can also be described by the following state-space form

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{7}$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $x(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix}$.

We assume that the target will either remain static (i.e., a fixed position in the environment) or move with constant speed, hence it can be model by the following equations

$$\dot{r_p}(t) = r_v(t), \tag{8}$$
$$\dot{r_v}(t) = 0, \tag{9}$$

where r_p and r_v are the position and velocity of the target. The state-space model of the target can be given as follows

$$\dot{r}(t) = Ar(t), \tag{10}$$

where $r(t) = \begin{bmatrix} r_p(t) \\ r_v(t) \end{bmatrix}$.

In order to track the target precisely during the mission, a suboptimal control law is designed for the leader. Let the state tracking error

$$e(t) = x(t) - r(t) \tag{11}$$

and the initial tracking error $e(0) = e_0$. Since the pair (A, B) is stabilisable, for given weighting constants $Q > 0$ and $R > 0$, there exists a positive definite P satisfying

$$A^T P + P A - P B R^{-1} B^T P + Q < 0, \tag{12}$$
$$e_0^T P e_0 < \gamma, \tag{13}$$

where $\gamma > 0$ is a positive gain. Motivated by [41, 42], we can then design the following feedback control protocol for the leader

$$u(t) = -R^{-1} B^T P e(t). \tag{14}$$

Substituting Eqs. 7 and 14 into the derivative of Eq. 11, the expressions for the closed-loop system dynamics can be obtained as

$$\dot{e}(t) = (A - B R^{-1} B^T P)e(t), \tag{15}$$

Since P satisfies Eq. 12, we have

$$(A - B R^{-1} B^T P)^T P + x P(A - B R^{-1} B^T P)$$
$$+ Q + P B R^{-1} B^T P < 0, \tag{16}$$

which implies that $A - B R^{-1} B^T P$ is Hurwitz, i.e., the closed-loop system Eq. 15 is asymptotically stable. Therefore, the corresponding cost is finite and given by

$$J = \int_0^\infty e^T (Q + P B R^{-1} B^T P)e \, dt. \tag{17}$$

Considering that Eq. 13 holds, we can easily find that $J < \gamma$ is satisfied, which means the suboptimality of the target tracking algorithm can be guaranteed.

Therefore, by using the control law Eq. 14, the leader robot can track the goal position within finite time and the performance of the closed-loop system can be tuned by adjusting Q, R and γ respectively. This coordination algorithm of the leader lays the foundation of steering the whole robot swarm to the desired target position.

Algorithm 2 Multiple goals tracking algorithm.

//initialisation
$Goal_N = 0$
while $\|e(t)\| \leq 0.1$ **do**
    if *Condition is met* & $Goal_N < Goal_{max}$ **then**
        | $Goal_N \leftarrow Goal_N + 1$
    end
end

With Algorithm 2, the swarm is able to follow the leader and tracks each goal in order. As shown in Algorithm 2, when the leader reaches a goal, the index of the current

goal increases one and the leader starts to reach the next destination until it reaches the final goal.

3 Experimental setup

3.1 General foundation

In this work, Webots [43] is chosen as the simulation platform. It provides a realistic simulation to design a customised environments. The software includes libraries for various robots and objects, which are provided in toolboxes. As a professional robotic platform, Webots has integrated cross-compilation systems allowing users to compile and upload the controllers to real-life robots with minimum modification, which facilitates the application for future work.

In Webots software, we used a number of TurtleBot [44] robots to investigate the proposed leader-follower mechanism. TurtleBot is an affordable, open-source mobile robot with excellent expandability. The miniature size of Turtle-Bot, 138 mm×178 mm, is ideal for research in swarm robotics. Created in 2010, TurtleBot is a popular platform and has been utilised in many swarm and multi robotic studies [45, 46]. As a mobile robot, TurtleBot is actuated by two differential wheels. Its maximum forward velocity is 0.22 m/s and maximum angular velocity is 2.84 rad/s. TurtleBot is equipped with a netbook, sensors and a gyroscope. 360 laser distance sensor and IMU are suitable equipment for the robotic tasks, such as mapping and localisation. With maximum payload of 15 kg, TurtleBot is able to be adopted for various tasks in the real world scenarios.

3.2 Swarm robotics experiments

In this section, we aimed to apply the aforementioned flocking mechanism for steering a swarm of mobile robots. In initialisation step at the beginning of each experiment, the orientation of each robot was randomly defined, $\theta \in [-\pi, +\pi]$, and the inter-robot distances with their neighbours were set within 0.3 m. In order to test the collective behaviour with different kinds of initial forms, we run experiments with four robots in different initialised shapes. In Fig. 1, a and c are the two different initialisation forms. (b) and (c) are the form of the swarm (a) and (c) correspondingly after a while ($t < 10$ s). In particular examples, both of the experiments took less than 10 s to achieve steady state. Since we evaluated the group performance of the teams' steady state, the initialised form does not play an important role in our experiments.

To apply the swarm mechanism, each robot has an individual controller which is implemented by its own
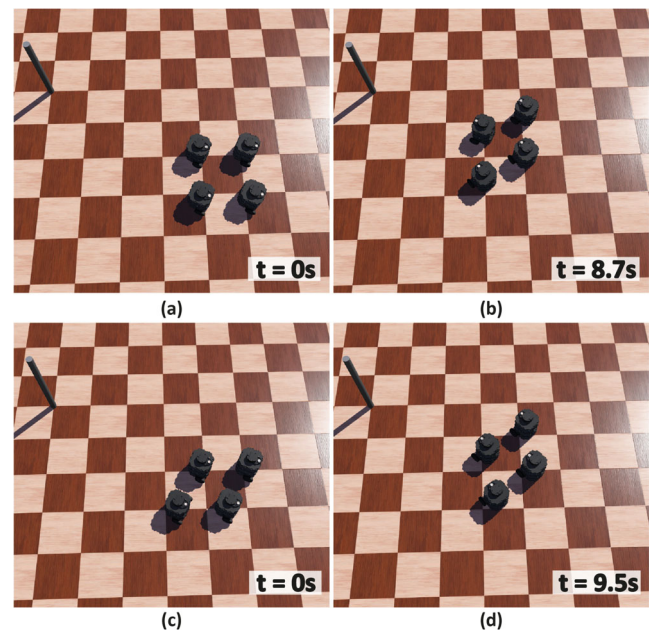

Fig. 1 Examples of preliminary tests for the robot group in different scenarios: **a** and **b** are the screenshots for the swarm with initialisation in a square. **c** and **d** are the screenshots for the swarm with initialisation in a parallelogram

microcontroller. According to the swarm mechanism, there are two forces that mainly affect the collective motion. One is the collective force, $\vec{F}_i$, and the other is the repulsive force, $\vec{F}_o$. The collective force depends on the relative positions of the neighbours, which can be calculated by summing the errors between neighbouring vector, r_{ij}, and desired distances, l_{ij}. Figure 2 shows an example of the initialisation of 7 robots in a swarm. Figure 2b illustrates $\vec{F}_i$ between robots. Apart from the leader, all robots are exerted forces from their neighbours. Compared with Fig. 2a, It is clear that forces have contributed to creating the desired distances between robots and the interaction network structure is more obvious.

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 surrounding neighbours in each robot's coordinate system. 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. 4. The component of the total force in the vertical direction was calculated in the same way using $\hat{n}_i^\perp$.

The repulsive force is calculated in a similar way. Prior to applying the repulsive force, the coordinate transformation was adopted because $\vec{F}_o$ is in the global coordinate frame, while the set of $\vec{F}_i$ controller on the TurtleBot is in the robot coordinate frame. Collective forces have weight parameters k which influence the forces that are applied to the robot,
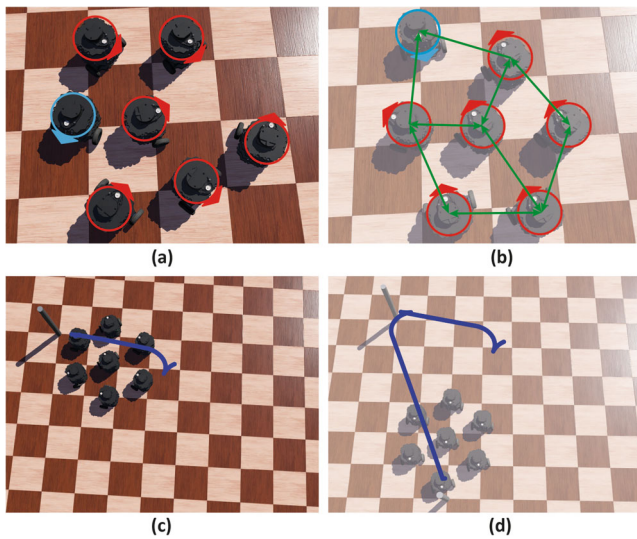
Fig. 2 **a** Initialisation of swarm; every robot is marked in a circle with a triangle. The triangle points along the robot's heading direction. The blue circle indicates the leader. **b** Force between robots; if there is a force between two connected robots, a green force arrow is used to represent the direction of the forces. **c** Leader-follower tracking; after adjustment between robots, the swarm is heading to the first goal by following the leader. The blue curve shows the trajectory of the leader. **d** Group motion; the swarm collectively move to multiple goals. The blue curve is the trajectory of the leader

for example, an increase in k leads to a bigger force to pull the swarm toward a direction and vice versa.

With these force information, the controllers start to calculate total forces and correspondingly change the robot kinematics. The transformation from force to the robot's kinematics is described by the following equations. $\vec{F}_i$ and $\vec{F}_g$ are related to forward velocity, $\hat{v}_i$, and angular velocity, w_i, which are two variables to describe the kinematics of the robot.

$$\|\hat{v}_i\| = m \|\vec{F}_i + \vec{F}_g\|, \tag{18}$$
$$w_i = n\angle(\vec{F}_i + \vec{F}_g), \tag{19}$$

where parameters m and n are tuned empirically to make sure each robot rapidly adjust its speed and heading direction, which is the fundamental of group behaviour.

The transformation between the goal position and the angular velocity of wheels can be derived as:

$$\begin{bmatrix} \|\hat{v}_i\| \\ w_i \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix}. \tag{20}$$

Positions and orientations of the robots are recorded at each sampling time. Figure 2c and d shows the trajectory of the leader achieving the first and second goals, respectively,

during the experiments. The swarm is able to follow the leader in achieving multiple goals.

3.3 Leader test

In this work, we assume that the target position of the whole robot swarm is only assigned to the leader, so it is a partially informed heterogeneous system that only leader knows about the target. Based on the relative position information between the target and the leader, the position tracking error $e(t)$ can be detected by the leader at each time instant, t. Then, by using proposed optimal control law Eq. 14, the leader will reach the desired location autonomously within finite time. Note that the tracking speed of the leader can be adjusted by tuning Q, R and γ in solving Eqs. 12 and 13, respectively. However, due to the physical constraints of the motors in TurtleBot, we cannot select arbitrary large tuning parameters. Besides, in order to generate a smooth trajectory for the leader, we should also set a proper maximum angular velocity in the feedback linearisation controller, which is implemented in the robot's low-level control design. Since the leader robot can be observed and tracked by at least one of the follower, when the leader is moving towards the target position, the connected followers can be steered accordingly. This guarantees the completion of the flocking tasks.

3.4 Metrics

The expected flocking performance is the group of robots collectively move along the track of the leader. To evaluate the group motion, we used two metrics which have commonly used in swarm robotics area [25, 29, 47, 48]: d_s and ξ. These two metrics were calculated and recorded at each sampling time during every experiment.

1. d_s is the average distance of the swarm, which is calculated with the following equation. By comparing the average distance of different groups, coherency and density can be evaluated. Typically, steady and relatively small values mean that the swarm has expected coherency.

$$d_s = \frac{2\sum_{i=1}^{N-1}\sum_{j=1}^{N}\|\vec{r}_{ij}\|}{N(N-1)}, \tag{21}$$

where N is the number of robots in the swarm and $\|\vec{r}_{ij}\|$ is the distance between robots i and j.

2. Apart from distance, swarm alignment is also critical measurements for evaluating the alignment of a group. We calculate swarm alignment as Eq. 22 in this paper. swarm alignment, ξ, indicates the agreement of the

robots' heading, which is related to the vectorial sum of each individual's rotation, shown as

$$\xi = \frac{\| \sum\limits_{i=1}^{N} \hat{n}_i \|}{N} \, , \tag{22}$$

where $\hat{n}_i$ is a unit vector along with the heading of robot i. ξ is between 0 and 1. Bigger value shows the better agreement of the robots' heading, in contrast, smaller one demonstrates a more disswarm alignmented heading performance.

Since the robots were initialised at random positions with randomly selected orientations, the values of the metrics are different between experiments at the beginning, $t = 0$ s.

In the following section, the results of experiments are presented and investigated to compare the impact of different parameters in the mechanism, Eq. 4. The parameters are related to time, population size ($N \in \{4, 7, 10\}$ robots), force ($k \in \{0.5, 1, 1.5\}$) and desired distance ($l_i j \in \{0.25, 0.3, 0.35\}$ m) between the robots. Each set of experiments has the same parameter setting and was repeated 10 times.
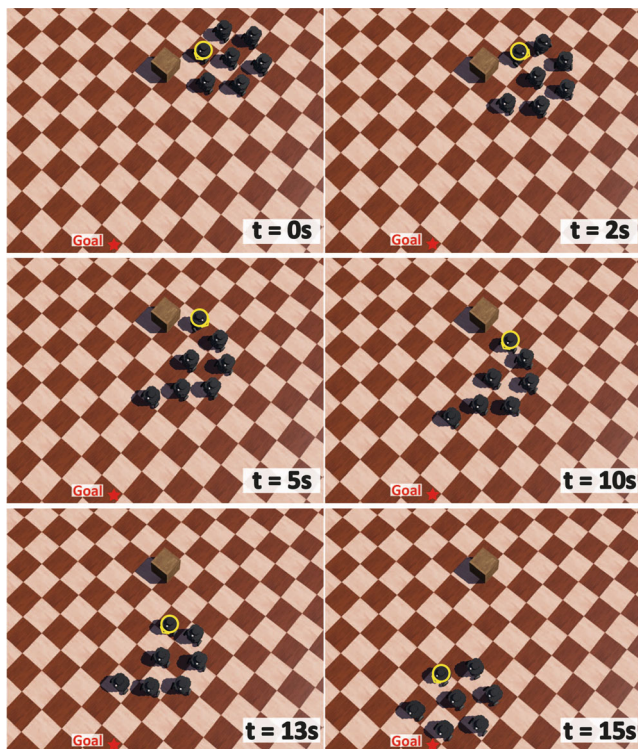
Fig. 3 A randomly selected example of 7 robots avoiding an obstacle while performing swarm flocking. The robot marked with yellow colour is the robot that detected an obstacle

4 Results & discussion

To address the influence of factors including time, population size, desired distance and force, various experiments were designed with independent object factors and fixed irrelevant variables. Each set of experiments was repeated 10 times to generate the following results.

Figure 3 is a randomly selected experiment where a group of robots are avoiding an obstacle. When a robot (marked with yellow circle) detected the obstacle which is on its way to the goal, the robot tried to change its heading direction because of the force from the obstacle. However, there were a few neighbours on its changing direction. Owning to the force from neighbours, it is obvious the robot was waiting from $t = 2$ s to $t = 5$ s. The robot started to chase the swarm from $t = 5$ s and ended up with joining the swarm at around 15 s.

4.1 Effects of time

To check the change of metrics during experiments, we implemented an experiment which corresponds to the Fig. 2c. Force parameter and desired distance were set as $k = 1, l_{ij} = 0.3$ m in a swarm with $N = 7$ robots.

Figure 4 shows the results of d_s and ξ during 22 s. After initialisation, the robots adjusted to keep appropriate distances from their neighbours in the first 10 s, while the leader headed to the goal from the beginning. From $t = 10$ s to $t = 16$ s, the swarm travelled to the goal by following the leader. As shown in Fig. 2c, trajectory of the leader robot
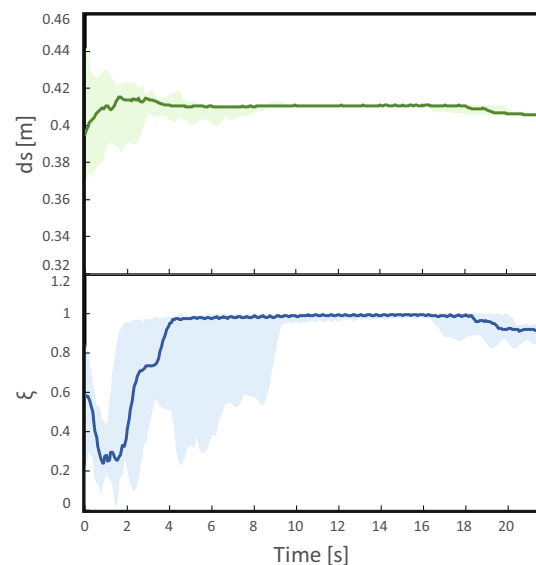
Fig. 4 Results of distance and swarm alignment during experiments with $k = 1$, $l_{ij} = 0.3$ with $N = 7$ robots within $t = 21$ s. The shaded area indicts the distribution of results and lines show the median of results

was almost a straight line. Both average distance d_s and swarm alignment ξ achieved and held steadily as expected value in this period. The swarm arrived the goal around $t = 16$ s. The leader started to turn and move to the next goal, which resulted in the slight change of both metrics.

Based on the upper graph of Fig. 4, the maximum and minimum d_s (maximum fluctuation) occurred at the beginning of the experiment, with approximately 0.44 m and 0.375 m. Then, the average distance maintained around 0.41 m.

Similarly, swarm alignment ξ changed dramatically at the first 10 s, with minimum value 0.02. During the travelling period, the swarm alignment almost stayed at 1, which indicates the swarm was able to maintain the coherency within this period of time. Turning triggers the adjustment between robots, which is the reason of the slight fluctuation after $t=16$ s.

4.2 Effects of population

With $k = 1$ and $l_{ij} = 0.3$, experiments were conducted using various population sizes $N \in \{4, 7, 10\}$ robots. Results were collected during the period when all experiments reach steady state, so the length of the data set might different among Sections 4.2, 4.3 and 4.4. Figure 5 shows the obtained results in different populations.

As shown in Fig. 5a, d_s rises as the population increases. The results of average distance were more consistent (less fluctuations) for the groups with $N=4$ and $N=7$ robots than $N=10$ robots. The results with $N \in \{4, 7\}$ robots were mainly distributed around $d_s=0.33$ m and $d_s=0.41$ m, respectively. There was enormous fluctuation for the results obtained with $N=10$ robots, which were distributed between $d_s=0.2$ m and $d_s=1.0$ m with median of $d_s=0.8$ m. A similar behaviour in relationship between population
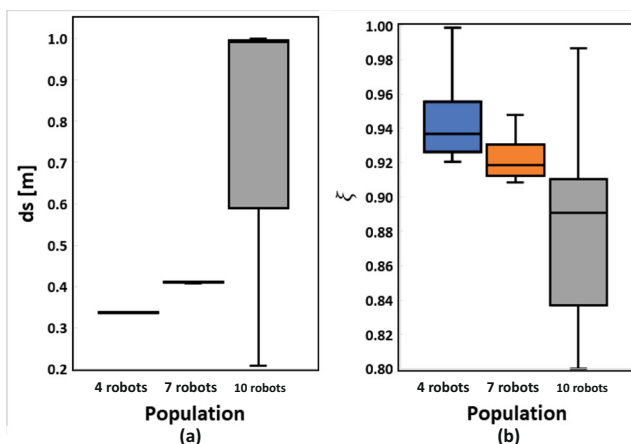
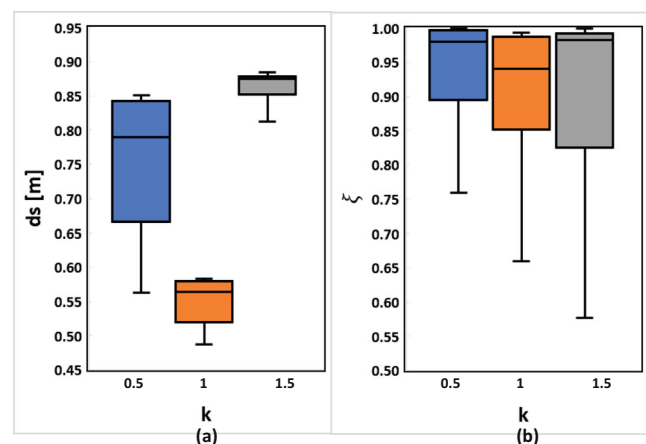size and coherency of the swarm was also reported in [12].

In terms of the swarm alignment ξ, increasing population triggers worse alignment. Similar results were also reported in [25]. The results from experiments with $N=7$ robots showed the tightest distribution, with approximately $\xi=0.92$. Experiments with $N=10$ robots have the worst group performance among these experiments and spread out more dramatically than the other two settings, with the median value around $\xi=0.89$. The obtained results from experiments with $N=4$ robots has the best distribution, with 0.94 and nearly 1 as the median value and maximum value respectively.

4.3 Effects of force

Figure 6 reveals the average distance and swarm alignment with $l_{ij} = 0.3$, $N=7$ robots and varying force parameters $k \in \{0.5, 1, 1.5\}$. According to Eq. 4, k is the parameter that relies on the strength of force $\vec{F}_i$.

Compared the distribution of desired distance in Fig. 6a, the swarm mechanism with $k = 1$ has the best coherency among the experiments. The average distance changed between 0.47 m to 0.58 m. It can be clearly observed that mechanism with $k = 0.5$ yields better coherency than the one with $k = 1.5$, but has a greater chance to change. It means the swarm with $k = 0.5$ tends to change the form compared with the other one.

Considering the results of Fig. 6b, all experiments have chance to reach a bigger value. The mechanism with $k = 1.5$ has greater variability in changing orientation, which leads to worse swarm alignment of the swarm. In contrast, $k = 0.5$ contributes the best swarm alignment among the experiments, with less dispersion of the data and the relatively big swarm alignment value.

Fig. 5 Results of average distance and swarm alignment during experiments with $k = 1$, $l_{ij} = 0.3$ using $N \in \{4, 7, 10\}$ robots. Blue, orange and grey boxes are for 4, 7 and 10 robots, respectively

Fig. 6 Results of average distance and swarm alignment during experiments with $l_{ij} = 0.3$, $N = 7$ with $k \in \{0.5, 1, 1.5\}$. Blue, orange and grey boxes are for $k = 0.5$, 1 and 1.5, respectively

4.4 Effects of desired distance

Desired distance is a parameter which involves with the expected distance between robots and their neighbours. Robots adjust their positions based on the difference between the estimated neighbouring distance and the expected distance. The impacts of different l_{ij} are illustrated in Fig. 7.

Shown in Fig. 7a, an increase in l_{ij} increases the average distance as expected. In case of $l_{ij} = 0.25$, it increases the risk of dispersive distribution of average distance, though it has a greater chance to reach the minimum value. The average distance of swarm with $l_{ij} = 0.3$ maintains around 0.41 m, which is enough for followers to track the leader. With 0.455 m, the coherency of the swarm using $l_{ij} = 0.35$ is the worst among the experiments, namely the larger desired distance might hinder the collective dynamics.

According to Fig. 7b, the swarm alignment has the chance to reach the maximum value with various settings of desired distances. A clear indication of the case with $l_{ij} = 0.25$ shows the observed results from the swarm performance spread out dramatically. In case of $l_{ij} = 0.35$, the swarm provided the best swarm alignment, although the results had greater distribution in comparison with other two settings.

The aforementioned analysis demonstrated the impact of population, force and desired distance on the group performance and also provided a guidance to tune the relevant parameters based on the desired swarm behaviour. According to the results and discussions given in this section, it can be concluded that the collision-free flocking task can be achieved by the proposed mechanism and the performance can also be potentially adjusted to fulfill the requirements of real-world applications. In terms of

Fig. 7 Results of average distance and swarm alignment during experiments with $N = 7$ robots and $k = 1$ with $l_{ij} \in \{0.25, 0.3, 0.35\}$ m. Blue, orange and grey boxes are for $k = 0.25, 0.3$ and 0.35, respectively

improving the swarm performance, methods such as neural collaborative filtering and fuzzy clustering algorithm [49] could be exploited to achieve group flocking behaviour of heterogeneous robot swarms.

4.5 Discussion

Previous work by Ferrante et al. [30] has established that the swarm collectively move to a goal direction in open spaces without considering the obstacles. On the contrary, the mechanism we proposed here can deal with such obstacles in the environment. The swarm is able to travel along the expected track and avoid obstacles. Even robots spread out when avoiding collision, the swarm can rebuild the network and continue to drive to the goal. Compared with a homogeneous mechanism, the experimental work presented here provided a solution which has less requirement for robot's hardware especially in terms of communication and storage. In previous work [50], apart from obtaining data from neighbouring robots, each robot in the swarm must know the location of the goal. On the contrary, in this study, only the leader robot is assumed to know the goal information and all follower robots just need to use the local information from the neighbours. In this case, the proposed solution can decrease the communication costs of individual robots and further make the swarm more affordable.

It is now well established from a seminal work by Şahin [51], each robot in a swarm system makes decisions only based on its neighbours' states and only interacts with its direct neighbours within a specific range, *sensing radius*, due to the limitations of real robots' hardware in practical scenarios. Therefore, direct communication in a swarm system without having an extra observer, *central controller*, is one of the challenges of implementing swarm scenarios using real mobile robots [20].

Some research studies [13, 29, 52, 53] relied on the acquisition of the location of each robot from an external observer, whereas some [30, 54] regarded each robot as an abstract particle without considering physical properties e.g. weight, size, motors speed and sensing 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 this work, the swarm is not controlled by a central controller. Every follower makes decisions based on its neighbours and surrounding obstacles, while the leader only requires the information of the goal and surrounding obstacles. Compared to those aforementioned works, the mechanism we proposed here is more realistic for the application in real robots.

In terms of the limitation of this work, we did not considered optimisation of the parameters in Eqs. 1–4.
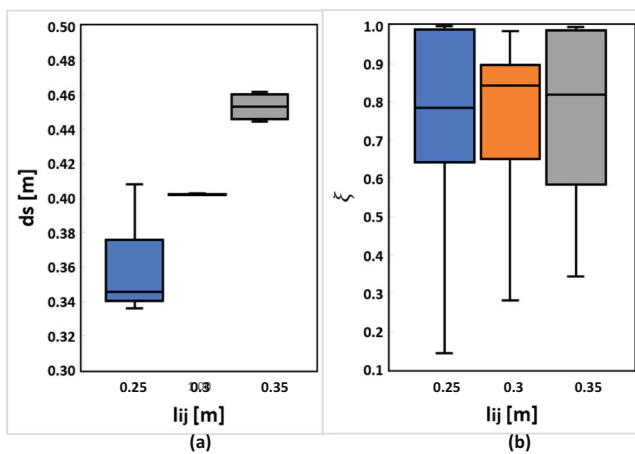
Hence, using optimisation algorithms, e.g. [55, 56], will benefit the choice of these parameters.

5 Conclusion

The main goal of this study was to design a generalised swarm mechanism which has the ability to adapt to a complex environment. Using a leader and a number of followers, the swarm network successfully finished the designed tasks, which has gone some way towards enhancing our understanding of the swarm application in the real world, such as, collective mapping and collective sensing. Compared with the previous work, one of the most obvious improvement to emerge from this study is that the swarm network has less requirement for power and storage. The analysis of factors undertaken here can be considered the first step from an abstract mechanism to the progression with real robots. In the future, a greater focus on application in real robots could produce findings that account more for precision agriculture, collective sensing and so on.

References

1. Camazine S, Deneubourg JL, Franks NR, Sneyd J, Bonabeau E, Theraula G (2003) Self-organization in biological systems. Princeton University Press, Princeton
2. Hein AM, Gil MA, Twomey CR, Couzin ID, Levin SA (2018) Conserved behavioral circuits govern high-speed decision-making in wild fish shoals. Proc Nat Acad Sci 115(48):12,224–12,228
3. Flack A, Nagy M, Fiedler W, Couzin ID, Wikelski M (2018) From local collective behavior to global migratory patterns in white storks. Science 360(6391):911–914
4. Torney CJ, Hopcraft JGC, Morrison TA, Couzin ID, Levin SA (2018) From single steps to mass migration: the problem of scale in the movement ecology of the serengeti wildebeest. Philosop Trans R Soc B Biol Sci 373(1746):20170,012
5. Kaiser A, Snezhko A, Aranson IS (2017) Flocking ferromagnetic colloids. Sci Adv 3(2):e1601,469
6. Hu J, Bhowmick P, Jang I, Arvin F, Lanzon A (2021) A decentralized cluster formation containment framework for multirobot systems. IEEE Trans Robot. https://doi.org/10.1109/TRO.2021.3071615
7. Schranz M, Di Caro GA, Schmickl T, Elmenreich W, Arvin F, Şekercioğlu A, Sende M (2021) Swarm intelligence and cyber-physical systems: concepts, challenges and future trends. Swarm Evolut Comput 60(100):762
8. Miletitch R, Dorigo M, Trianni V (2018) Balancing exploitation of renewable resources by a robot swarm. Swarm Intell 12(4):307–326
9. Hu J, Turgut AE, Krajník T, Lennox B, Arvin F (2020) Occlusion-based coordination protocol design for autonomous robotic shepherding tasks. IEEE Trans Cognitive Develop Syst. https://doi.org/10.1109/TCDS.2020.3018549
10. Tarapore D, Timmis J, Christensen AL (2019) Fault detection in a swarm of physical robots based on behavioral outlier detection. IEEE Trans Robot 35(6):1516–1522
11. Hu J, Niu H, Carrasco J, Lennox B, Arvin F (2020) Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. IEEE Trans Vehic Technol 69(12):14,413–14,423
12. Amjadi AS, Raoufi M, Turgut AE, Broughton G, Krajník T, Arvin F (2019) Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation. arXiv:1907.09585
13. Hu J, Bhowmick P, Arvin F, Lanzon A, Lennox B (2020) Cooperative control of heterogeneous connected vehicle platoons: an adaptive leader-following approach. IEEE Robot Autom Lett 5(2):977–984
14. Grover P, Bakshi K, Theodorou EA (2018) A mean-field game model for homogeneous flocking. Chaos An Interdiscipli J Nonlinear Sci 28(6):061,103
15. Wong WK, Ye S, Liu H, Wang Y (2020) Effective mobile target searching using robots. Mob Netw Appl 1–17
16. Liu Z, West C, Lennox B, Arvin F (2020) Local bearing estimation for a swarm of low-cost miniature robots. Sensors 20(11)
17. Hu J, Turgut AE, Lennox B, Arvin F (2021) Robust formation coordination of robot swarms with nonlinear dynamics and unknown disturbances: Design and experiments. IEEE Transactions on Circuits and Systems II, Express Briefs. https://doi.org/10.1109/TCSII.2021.3074705
18. He C, Feng Z, Ren Z (2018) A flocking algorithm for multi-agent systems with connectivity preservation under hybrid metric-topological interactions. PloS one 13(2):e0192,987
19. Thrun MC (2018) Projection-based clustering through self-organization and swarm intelligence: combining cluster analysis with the visualization of high-dimensional data. Springer, Berlin
20. Dorigo M, Theraulaz G, Trianni V (2020) Reflections on the future of swarm robotics. Sci Robot 5(49)
21. Wu K, Hu J, Lennox B, Arvin F (2021) SDP-based robust formation-containment coordination of swarm robotic systems with input saturation. J Intell Robot Syst 102(12)
22. Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp 25–34
23. Hayes AT, Dormiani-Tabatabaei P (2002) Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In: IEEE International conference on robotics and automation, vol 4, pp 3900–3905
24. Campo A, Nouyan S, Birattari M, Groß R, Dorigo M (2006) Negotiation of goal direction for cooperative transport. In: International workshop on ant colony optimization and swarm intelligence. Springer, Berlin, pp 191–202

25. Turgut AE, Çelikkanat H, Gökçe F, Şahin E (2008) Self-organized flocking in mobile robot swarms. Swarm Intell 2(2-4):97–120

26. Li B, Wu ZX, Guan JY (2019) Collective motion patterns of self-propelled agents with both velocity alignment and aggregation interactions. Phys Rev E 99(2):022,609

27. Hiraiwa T (2019) Two types of exclusion interactions for self-propelled objects and collective motion induced by their combination. Phys Rev E 99(1):012,614

28. Ferrante E, Turgut AE, Huepe C, Stranieri A, Pinciroli C, Dorigo M (2012) Self-organized flocking with a mobile robot swarm: a novel motion control method. Adapt Behav 20(6):460–477

29. Na S, Qiu Y, Turgut AE, Ulrich J, Krajník T, Yue S, Lennox B, Arvin F (2020) Bio-inspired artificial pheromone system for swarm robotics applications. Adaptive Behav 1059712320918936

30. Ferrante E, Turgut AE, Dorigo M, Huepe C (2013) Collective motion dynamics of active solids and active crystals. New J Phys 15(9):095,011

31. Ni W, Cheng D (2010) Leader-following consensus of multi-agent systems under fixed and switching topologies. Syst Control Lett 59(3-4):209–217

32. Hu J, Bhowmick P (2020) A consensus-based robust secondary voltage and frequency control scheme for islanded microgrids. Int J Electric Power Energ Syst 116(105):575

33. Zhu W, Cheng D (2010) Leader-following consensus of second-order agents with multiple time-varying delays. Automatica 46(12):1994–1999

34. Wu K, Hu J, Lennox B, Arvin F (2021) Finite-time bearing-only formation tracking of heterogeneous mobile robots with collision avoidance. IEEE Transactions on Circuits and Systems II, Express Briefs. https://doi.org/10.1109/TCSII.2021.3066555

35. Hu J, Bhowmick P, Lanzon A (2020) Two-layer distributed formation-containment control strategy for linear swarm systems: Algorithm and experiments. Int J Robust Nonlinear Control 30(16):6433–6453

36. Bhavana T, Nithya M, Rajesh M (2017) Leader-follower co-ordination of multiple robots with obstacle avoidance. In: 2017 International conference on smart technologies for smart nation (smarttechcon). IEEE, pp 613–617

37. Lee G, Chwa D (2018) Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. Intel Serv Robot 11(1):127–138

38. Wu X, Wang S, Xing M (2018) Observer-based leader-following formation control for multi-robot with obstacle avoidance. IEEE Access 7:14,791–14,798

39. Ren W, Atkins E (2007) Distributed multi-vehicle coordinated control via local information exchange. Int J Robust Nonlinear Control 17(10-11):1002–1033

40. Hu J, Bhowmick P, Lanzon A (2020) Distributed adaptive time-varying group formation tracking for multi-agent systems with multiple leaders on directed graphs. IEEE Trans Control Netw Syst 7(1):140–150

41. Jiao J, Trentelman HL, Camlibel MK (2019) A suboptimality approach to distributed linear quadratic optimal control. IEEE Trans Autom Control 65(3):1218–1225

42. Trentelman HL, Stoorvogel AA, Hautus M (2012) Control theory for linear systems. Springer Science & Business Media, Berlin

43. Michel O (2004) Cyberbotics ltd. webots™: professional mobile robot simulation. Int J Adv Robot Syst 1(1):5

44. Garage W (2011) Turtlebot. Website: http://turtlebot.com/last visited 11–25

45. Gallardo N, Pai K, Erol BA, Benavidez P, Jamshidi M (2016) Formation control implementation using kobuki turtlebots and parrot bebop drone. In: 2016 World automation congress (WAC). IEEE, pp 1–6

46. Horton M, Chen L, Samanta B (2017) Enhancing the security of iot enabled robotics: Protecting turtlebot file system and communication. In: 2017 International conference on computing, networking and communications (ICNC). IEEE, pp 662–666

47. Vicsek T, Czirók A., Ben-Jacob E, Cohen I, Shochet O (1995) Novel type of phase transition in a system of self-driven particles. Phys Rev Lett 75(6):1226

48. Na S, Raoufi M, Turgut AE, Krajník T, Arvin F (2019) Extended artificial pheromone system for swarm robotic applications. In: Artificial life conference proceedings. MIT Press, pp 608–615

49. Gao H, Xu Y, Yin Y, Zhang W, Li R, Wang X (2019) Context-aware qos prediction with neural collaborative filtering for internet-of-things services. IEEE Int Things J 7(5):4532–4542

50. Ban Z, West C, Lennox B, Arvin F (2020) Self-organised flocking with simulated homogeneous robotic swarm. In: EAI International conference on collaborative computing

51. Şahin E. (2004) Swarm robotics: From sources of inspiration to domains of application. In: International workshop on swarm robotics. Springer, pp 10–20

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

53. Arvin F, Turgut AE, Krajník T, Rahimi S, Okay IE, Yue S, Watson S, Lennox B (2018) Φclust: Pheromone-based Aggregation for Robotic Swarms. In: 2018 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE, pp 4288–4294

54. Jia Y, Vicsek T (2019) Modelling hierarchical flocking. New J Phys 21(9):093,048

55. Sun G, Ma P, Ren J, Zhang A, Jia X (2018) A stability constrained adaptive alpha for gravitational search algorithm. Knowl-Based Syst 139:200–213

56. Zhang A, Sun G, Ren J, Li X, Wang Z, Jia X (2016) A dynamic neighborhood learning-based gravitational search algorithm. IEEE Trans Cybern 48(1):436–447