

# Morphogen Diffusion Algorithms for Tracking and Herding Using A Swarm of Kilobots

Hyondong Oh · Ataollah R. Shiraz ·  
Yaochu Jin

Received: date / Accepted: date

**Abstract** This paper investigates self-organised collective formation control using swarm robots. In particular, we focus on collective tracking and herding using a large number of very simple robots. To this end, we choose kilobots as our swarm robot testbed due to its low price and attractive operational scalability. Note, however, that kilobots have extremely limited locomotion, sensing and communication capabilities. To handle these limitations, a number of new control algorithms based on morphogen diffusion and network connectivity preservation have been suggested for collective object tracking and herding. Numerical simulations of large scale swarm systems as well as preliminary physical experiments with a relatively small number of kilobots have been performed to verify the effectiveness of the proposed algorithms.

**Keywords** Swarm robotics · object tracking · morphogen diffusion · network connectivity preservation · kilobots

## 1 Introduction

In recent years, due to the rapid advancements of robotics technology which can build a large number of simple and inexpensive robots, considerable interests have been raised in developing swarm pattern formation algorithms. Specific application for these swarming algorithms include, but not limited to, collaborative search and rescue, deployment of sensor networks, and collective

---

This work was funded by the European Commission 7th Framework Program, Project No. 601062, SWARM-ORGAN.

Hyondong Oh  
Department of Aeronautical and Automotive Engineering, Loughborough University,  
Loughborough, UK E-mail: h.oh@lboro.ac.uk

Ataollah R. Shiraz and Yaochu Jin  
Department of Computing, University of Surrey, Guildford, UK  
E-mail: {a.ramezanshirazi, yaochu.jin}@surrey.ac.uk

transportation and construction. Pattern formation in swarm robotic systems represents the coordination of a group of robots to maintain a formation with a certain shape of either a pre-defined pattern or adaptively formed pattern in a self-organised way with local interactions between robots and with the environment. In the former case, a group of robots are required to maintain a desired spatial pattern by individual robots aiming to keep a specific orientation and a relative distance to the others. In the latter case, global patterns emerge from local interactions between individual robots, and these emergent patterns can often be seen in natural systems, in particular in biological organisms and ecosystems such as patterning in seashells and fishes [1], and bird flocking, ant colony and fish schooling [2, 3].

Pattern formation algorithms for swarm robotic systems presented in the literature are developed primarily to be operated on a large collection of robots [4–7]. However, in reality, due to real-world constraints such as cost, time, and complexity of building and testing hardware systems, most research work has been performed in simulations only with an approximated model of sensors and robots, or experiments with a small number of robots at most. This approach may have several limitations. Firstly, it might be difficult to accurately model robots’ interaction with each other and with the environments, which can lead to a significant discrepancy between simulations and real experiments. In addition, unknown sensing noise and disturbance from environments as well as computation capability of a simple robot may seriously degrade the performance in experiments. Furthermore, experimental verification of algorithms designed for swarm behaviours with just a few robots may not be sufficient, as many issues such as scalability of the algorithm can be tested only in large-scale systems [8]. Thus, to rigorously validate swarm pattern formation algorithms, it is essential to test them on large-scale real robotic systems. Note that as there exist obvious trade-offs between the cost and the capability of the robot platforms such as movement and sensing capability [8–10], an adequate robotic test-bed needs to be carefully chosen depending on the requirements of a specific pattern formation algorithm.

In this study, keeping in mind that the developed algorithms will have to be tested in a large-scale (over a hundred) real robots ultimately, we select kilobots [8] as our swarm robot testbed with its low price and excellent operational scalability despite its limitations such as no self-localisation, no directional sensing and very limited communication capability. Considering these constraints of the kilobots, many existing pattern formation algorithms relying on the robot’s position and orientation information in a certain coordinate system [2, 4, 11–14] are difficult to be utilised. Since kilobots can sense only a distance or a light intensity level without any directionality, the direction of travel towards the other kilobots or a light source can only be obtained by performing random movements several times and observing the time variation of corresponding measurements; this makes it hard for the robots to track unknown moving objects or maintain a desired pattern. Lastly, due to constraints on data transmission rates between the robots, a high-level track-

ing and patterning algorithm needs to be based on simple local interactions in a decentralised way.

Among other collective behaviours to achieve swarm pattern formation, tracking randomly moving objects, surrounding and finally herding them into a pre-defined shape using biologically-inspired approaches is set to the final goal of this study. This objective might be described by the idea that a specific type of agent needs to find and contact the objects, but many others may be required to push the objects into a certain desired shape while maintaining the swarm group. Similarly, the entire swarm may need to be protected from the environment in much the same way that skin protects a biological organism. To cope with all constraints listed above for object tracking and herding, this paper proposes the use of a morphogen gradient diffusion process (found in biological pattern formation where the diffusion of protein concentration among cells determines polarity of cell and cell fate) and information on the number of neighbours (i.e. the neighbouring density) related to the network connectivity of the swarm group.

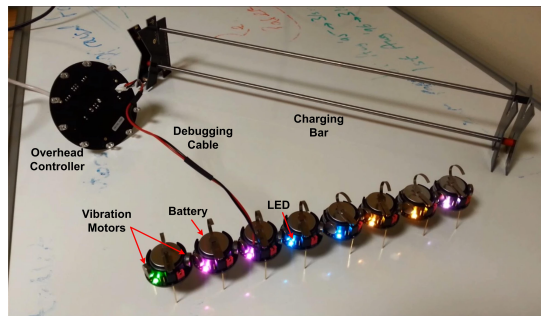
Even though similar approaches can be found in [5–7, 15], the key difference is that this study considers realistic constraints residing in the robot explicitly. For instance, Mamei *et al.* [5] presented a simple but very effective pattern formation algorithm without using directionality, but they assumed that robots can pass through each other without a physical size, which is difficult to be implemented in a real robot. Numerical simulations are performed in a *kbsim* (2D simulator designed for the kilobot with constraints in Python code by Halme [16]), and several preliminary experiments are performed with a smaller number of kilobots to check the feasibility of the proposed algorithm.

The remaining part of this paper is structured as follows. Hardware and technical specifications of kilobots and a kilobot simulator are introduced in Section 2. In Section 3, a detailed description on pattern formation algorithms using the concept of network connectivity and gradient diffusion is provided. Conclusions and future work are provided in Section 4.

## 2 Swarm Robotics Testbed

Kilobots, the swarm robot testbed we consider in this study, are developed by Rubenstein *et al.* [8] to address the issue of the swarm system size, pointing out that for technical reasons of hardware cost and complexity, most swarms contain a few tens of robots at most. To deal with swarms having higher numbers, they designed robot units made of cheap parts that are easy to assemble.

For locomotion, kilobots use two sealed coin shaped vibration motors with three sticks, adopting the slip-stick principle [17], as shown in Fig. 1. This allows both clockwise and counter-clockwise rotation as well as straight movement by differentially controlling the magnitude of vibration of two motors. The slip-stick locomotion significantly reduces the hardware complexity, robot size as well as cost compared to typical wheel drive robotic systems. The

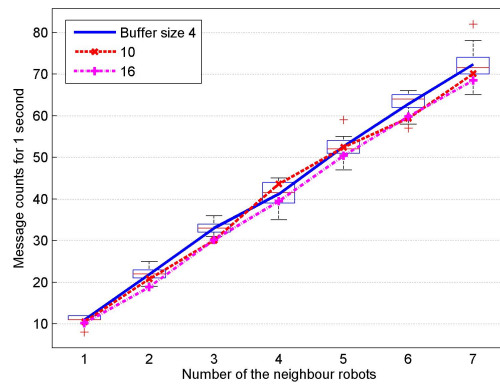


**Fig. 1** Kilobot platform including overhead controller and battery charging bar.

controller for the kilobots is an Atmega328 microprocessor which has 32K of memory and runs at 8 MHz. For communication, an infrared (IR) emitter and transmitter are used while allowing digital message transmission between robots at rates up to 30 kb/s within a range of 10 cm and a distance to a neighbouring robot is estimated by measuring the intensity of the incoming IR light. Current setting is sending a message at about every 0.1 second by interrupt and receiving process is triggered at the end of each byte coming. Figure 2 plots the received message counts (whenever receiving data from neighbouring robots, the count increases by one) of a kilobot for a fixed period of time, which shows the reliable communication property regardless of the received data buffer size. This message count can be used to determine the number of neighbouring robots without the use of an individual robot id. The kilobot also has an ambient light intensity sensor. Furthermore, in order to make the system scalable to a large number of robots, an overhead infrared transmitter is developed, so that the power can be switched on and off and the program can be uploaded remotely to all robots in the collective at the same time, regardless of the number of robots. Battery charging is also scalable by hanging the robots over a charging bar as shown in Fig. 1. More details of the kilobots can be found in [8].

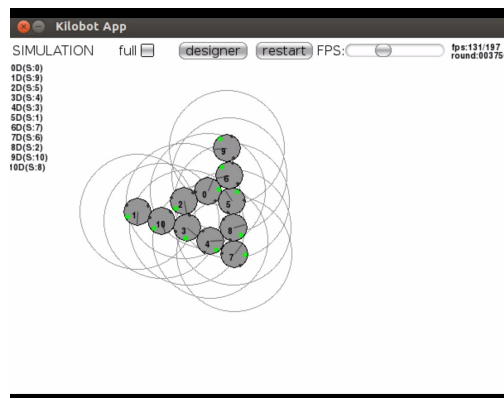
Note that although the kilobot testbed would make it possible to test swarm algorithms on collectives of robots of an order larger than that of existing systems, we need to carefully consider the limitation of the robot: IR communication with a few bytes up to 10 cm; distance and ambient light sensing without any directionality; no self-localisation capability (i.e. kilobots know neither their position nor orientation as there is no encoder/sensor to trace the movement or dead-reckoning); and inaccurate locomotion by vibrating motors sensitive to the surface and calibration settings. Besides, on starting a vibration motor, a spin up period is required by applying a high vibration power to overcome the friction and inertia; this often causes unpredictable slipping, bouncing off the surface, and control delays.

As feasibility study whether or not kilobots with such limited capabilities are able to reliably accomplish the aforementioned complex tracking and herding task, in this paper, the problem is simplified in that an object is static



**Fig. 2** Message counts for one second depending on the number of neighbour robots.

rather than dynamic in some cases, and an object tracking and herding process is not entirely integrated as yet. Moreover, for prototyping patterning algorithms on a swarm of kilobots before testing them on real robots, a kilobot simulator called *kbsim* is used at the current stage of the research, which is designed for simulating the kilobots in Python code with Pygame library for a user interface by Halme [16] as shown in Fig. 3. The *kbsim* simulates the most important functionality of the kilobots in terms of sensing, communication and movement, however, noise and disturbance from sensing, motor control and physical interactions between robots are not explicitly considered.



**Fig. 3** A sample snapshot of the kilobot simulator showing 'A' pattern formation [16].

### 3 Collective Behaviours for Object Tracking and Herding with Preserved Swarm Connectivity

This section introduces several pattern formation algorithms for object tracking and herding taking into account the capability of the kilobots based on light source tracking and morphogen diffusion.

#### 3.1 Object Tracking by Maintaining Network Connectivity

Providing that an initial swarm group is formed and part of the robots in the group is in the effective range of a light source as shown in Fig. 4(a), the objective in this section is: i) to approach to the light source (which can be considered tracking and surrounding a stationary target), ii) while maintaining group aggregation and the network connectivity with the group, which is termed as swarm coherence as in [7]. For this, spatiotemporal decision parameters are evaluated by each robot: the first one is  $\alpha = N_k - N_{k-1}$  as a temporal network connectivity measure at time instance  $k$  where  $N$  represents the number of neighbours, and the second one is  $\beta_k = N_k - \bar{N}_k$  as a spatial connectivity measure where  $\bar{N}$  represents the average number of neighbours' neighbours. By analysing these parameters, it can be inferred if each robot behaves correctly to maintain swarm coherence at the right place during the tracking. For instance, if  $\alpha$  is less than zero, it means that the robot is losing some of their network connections, or if  $\beta$  is less than zero, it means that the robot is either at the edge or sparse area in the group. In those cases, the robot is about to lose swarm coherence, and therefore a necessary action needs to be taken. To regain swarm coherence from this situation during an object tracking process, we take a similar approach presented in [7], which is turning backwards (180 degrees) immediately, and otherwise going towards a light source (target). This behaviour leads to the robots in the group maintaining swarm coherence as if they were attached with an elastic connection. The object tracking can be achieved by three behaviours depending on the network connectivity as:

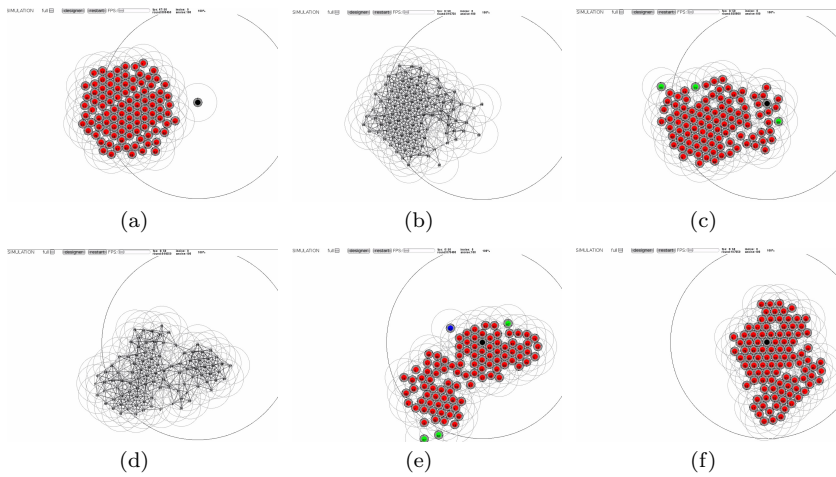
$$\begin{cases} \textit{Turn backwards}, & \text{if } N < N_{th}, \alpha \leq 0, \beta < \beta_{th} \\ \textit{Straight}, & \text{if } N < N_{th}, \alpha > 0, \beta < \beta_{th} \\ \textit{Towards target}, & \textit{otherwise}. \end{cases} \quad (1)$$

where the values  $N_{th}$  and  $\beta_{th}$  are positive threshold values for the number of neighbours and the spatial connectivity measure, respectively.

When a robot does not have enough neighbours ( $N < N_{th}$ ) and is at the edge of the group ( $\beta < \beta_{th}$ ) while losing network connections ( $\alpha \leq 0$ ), the robot should turn backwards to avoid losing swarm coherence. When the robot is in an unfavourable situation ( $N < N_{th}$  and  $\beta < \beta_{th}$ ) but gaining network connections ( $\alpha > 0$ ), the robot goes straight for a few time steps to gain more network connections and secure swarm coherence. Otherwise, the robot goes towards a target (the centre of a light source). The direction of travel towards a target is obtained by performing a few random movements until the

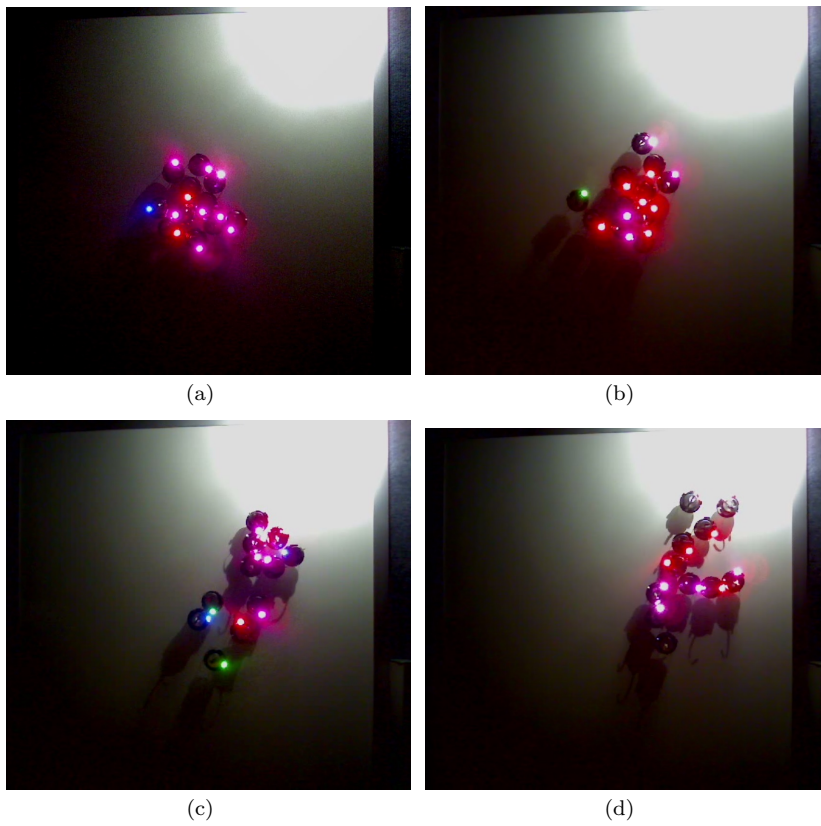
right direction is identified, and if the robot is out of light source range, it moves randomly. The benefit of this algorithm is that it requires the minimal exchange of information between robots, which is the number of neighbours  $N$  to accomplish the mission. Considering the locomotion of kilobots, ‘turning backwards’ behaviour is achieved by performing one direction turning for a certain number of time steps continuously. Note that ‘turning backwards’ behaviour does not necessarily need to be a precise 180-degree turn; instead, it is sufficient to make a turn to the opposite direction to maintain swarm coherence.

Figure 4 shows the sample process of target (light source) tracking. An initial robot group is given and the part of the group is within the effective range of the light source represented as a solid circle line. The robots outside the light source range initially move randomly, but as the remaining robots move towards the light source, they were attracted to the swarm by ‘turning backwards’ and ‘go straight’ behaviour. Throughout the process of Figs. 4(a)~(d), the entire robot swarm successfully moves towards the light source (the robot with a black dot) to surround it while maintaining swarm coherence. Moreover, a proof-of-concept experiment is performed using 11 kilobots as shown in Fig. 5, which demonstrates swarm movement towards the light source in spite of the ambient light sensing noise and inaccuracy.



**Fig. 4** Light source tracking process while maintaining network connectivity using the kilobot simulator with parameters  $(N_{th}, \beta_{th}) = (3, 2)$  (large circle: effective range of a light source, small circles: communication range, red: towards the target, blue: going straight, and green: turning backwards).

Lastly, Figure 6 shows the time history of the distance and the final distance (at iteration 18,000) between swarm group (40 kilobots) centroid and the target, depending on the threshold parameters on  $N_{th}$  and  $\beta_{th}$  in



**Fig. 5** Light source tracking using kilobots with parameters  $(N_{th}, \beta_{th}) = (3, 2)$ .

Eq. (1). From this figure, appropriate parameters can be identified as around  $(N_{th}, \beta_{th}) = (3, 1)$  since if ‘turning backwards’ is performed with too many or too few neighbours required ( $N_{th} = 1$  or  $5$  and  $\beta_{th} = -1$  or  $5$ ), it slows down the progress of the swarm movement towards the target with frequent turning backwards or leads to loss of network connectivity of some robots to the group. More detailed performance analysis in terms of performance metrics, e.g. those discussed in [18], will be followed as future work by changing the number of kilobots.

### 3.2 Object Tracking Using Morphogen Diffusion and Morphogen Gradient Following

This section utilises the morphogen diffusion process for object tracking. To maintain swarm coherence while tracking a moving object, each robot moves in a way that maintains their desired gradient concentration values. A morphogen gradient is initiated by a target object (or origin) by sending its neighbours



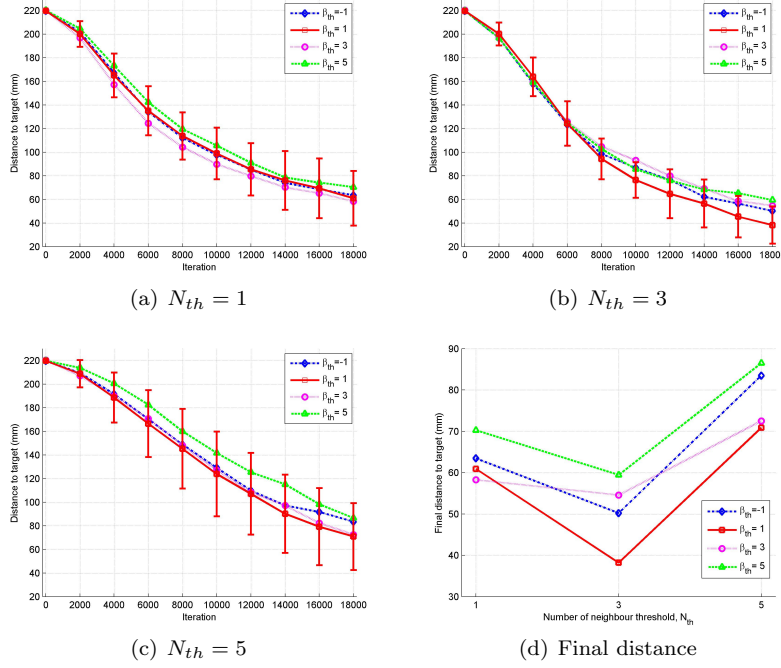


Fig. 6 Distance between group centroid and target averaged over 20 simulation runs.

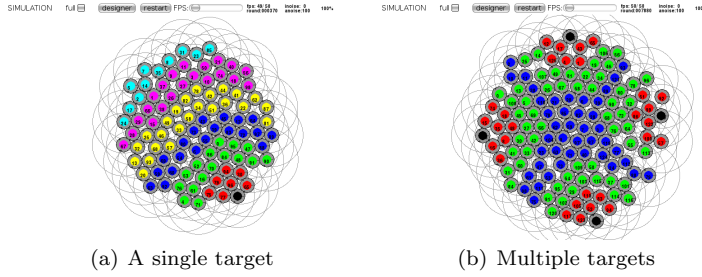


Fig. 7 Morphogen diffusion from the target (the robot with a black dot). Different colours represent different classes of concentration values.

a message with an initial value of  $h = 0$ . This gradient value  $h$  is forwarded outwards from the origin (target) incremented by one within a communication range of  $r$ . Each robot maintains the minimum gradient value (termed as concentration from now on) among received ones and ignores messages containing larger values, in order to prevent the gradient from backwardly diffusing. This gradient diffusion forms concentric circular rings of a width of approximately  $r$  and of different colours with corresponding concentration values around a target as shown in Fig. 7. It is worthwhile mentioning that the concentration

value can provide estimation of a straight line (or shortest) distance  $d$  of the robot from the target by the relation  $d \leq h \cdot r$ , which can then be improved by using an average neighbour density and smoothing algorithms, as explained in [6].

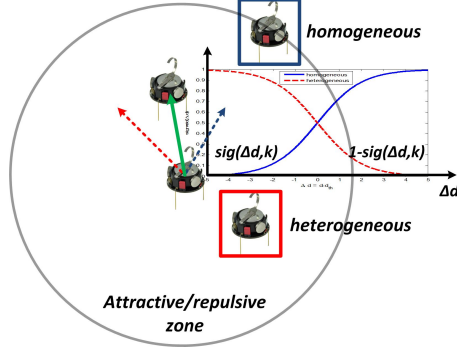


Fig. 8 Illustration of distance-based attraction/repulsion controller.

To make the robot follow other robots with the corresponding concentration value, a distance-based attraction/repulsion controller is used. Assuming that the robot at the centre is attracted to the homogeneous robot (blue) and repelled to the heterogeneous robot (red), as illustrated in Fig. 8, the net direction of travel is supposed to be where the green arrow points. However, as the kilobot has no directional sensing and no self-localisation capabilities, the movement direction is obtained indirectly by defining and analysing the following objective function for each robot:

$$f = w_a \text{sig}(\Delta d_a, k_a) + (1 - w_r \text{sig}(\Delta d_r, k_r)), \quad (2)$$

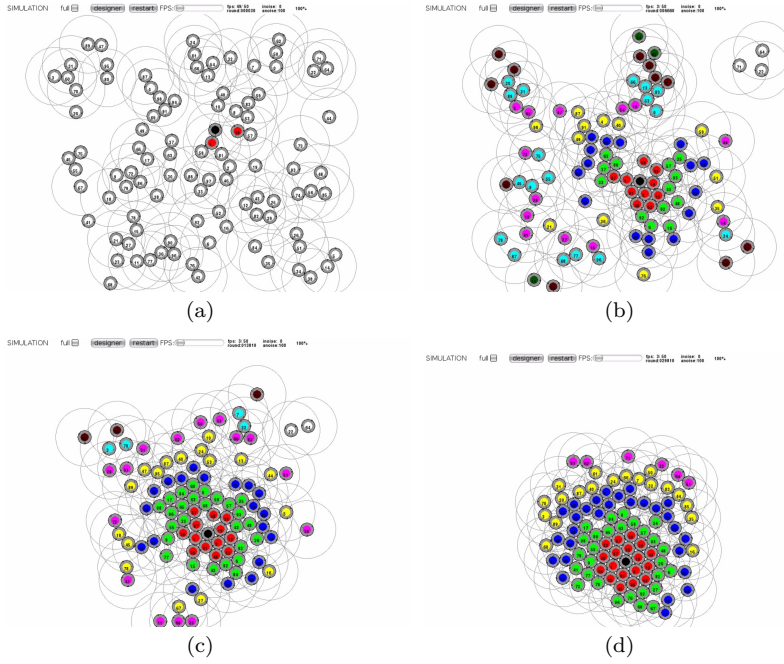
where

$$\Delta d_a = \sum_{j=1}^{N_a} \frac{d_j - d_{a,th}}{N_a}, \quad \Delta d_r = \sum_{j=1}^{N_r} \frac{d_j - d_{r,th}}{N_r}, \quad (3)$$

$$\text{sig}(x, k) = \frac{1}{1 + e^{-kx}}, \quad (4)$$

$k_a$  and  $k_r$  represent the slope of a sigmoid function,  $w_a$  and  $w_r$  are positive weighting factors,  $N_a$  and  $N_r$  are the number of robots to be attracted and repelled, respectively,  $d_j$  is the distance between the robot in concern and its neighbour robots.  $d_{a,th}$  and  $d_{r,th}$  are the threshold distance which determines an effective sensing range. From the above equation, it can be shown that the less the value  $f$  is, the better the robot is located for the attraction and repulsion mission. Thus, if the time derivative of the objective function is less than zero ( $df/dt < 0$ ), the robot keeps its previous movement direction, otherwise the robot moves randomly. If the objective of the robot is to be either attracted or repelled to the others, then only corresponding term in the

function  $f$  is used. From the following, two gradient following algorithms for object tracking are proposed depending on the use of concentration values.

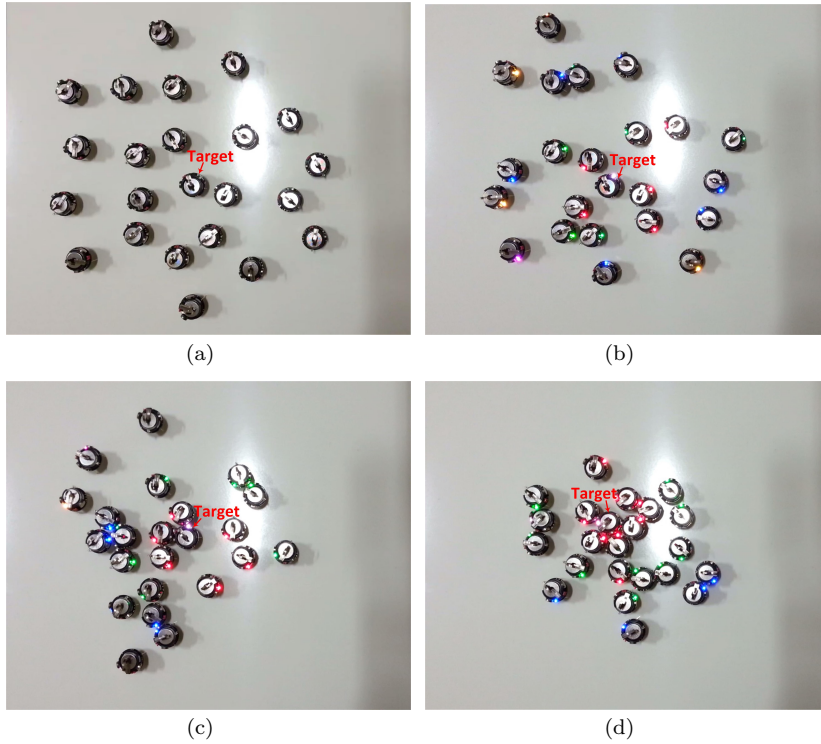


**Fig. 9** Process of surrounding a moving target by following robots having a lower concentration of morphogen with a distance-based attraction controller with parameters  $(d_{a,th}, k_a, w_a) = (69, 0.05, 128)$ .

### 3.2.1 Algorithm I: Following the morphogen gradient

In order to follow an object while maintaining swarm coherence, the robot is programmed to follow their preceding neighbours whose morphogen concentration is lower than its own using the attraction term in Eq. (2). By doing this, robots are attracted to each other as well as going towards the target with only local communication of morphogen gradient and distance between them. This allows a robot swarm to aggregate in one group around a target as long as the robot is connected to a target via multiple communication hops, and consequently to follow a moving target as a group. Figure 9 shows the process of tracking and surrounding a randomly moving target using morphogen diffusion and distance-based attraction controller with a kilobot simulator. At the beginning, the robots are randomly distributed, and a gradient value is propagated from a target (coloured as black) in the centre to the other robots. The robots coloured in white, which are not connected to the target through

morphogen diffusion, randomly move until they obtain a gradient value from their neighbours. The robots are progressively attracted towards the target by following their neighbours whose concentration is lower than its own, leading to surrounding the moving target successfully. Figure 10 shows a preliminary experiment result of the same process using 23 kilobots.



**Fig. 10** Experiment result of surrounding a moving target by following robots having a lower concentration with a distance-based attraction controller with parameters  $(d_{a,th}, k_a, w_a) = (80, 0.05, 128)$ .

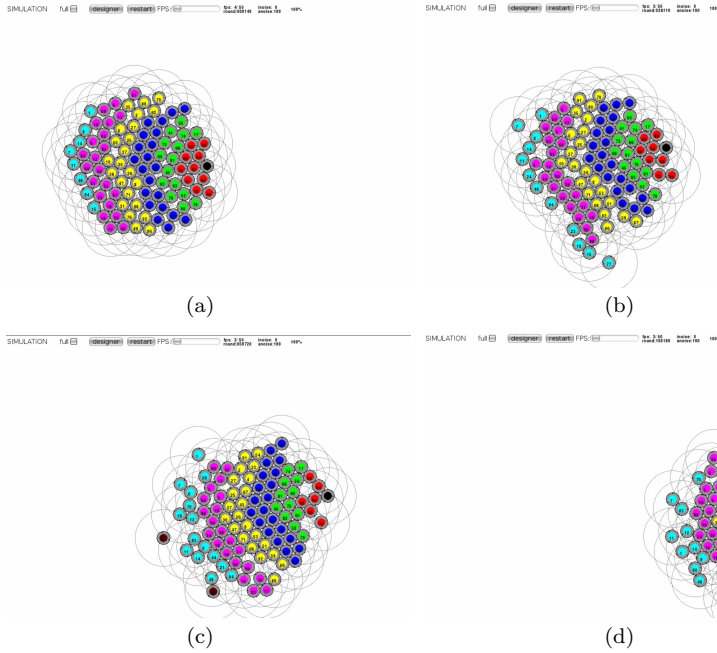
### 3.2.2 Algorithm II: Maintaining initial morphogen gradients for loose formation keeping

In the previous section, the objective of the robots is to follow the preceding robots only, where there are no other specific roles imposed on the robots. They can be anywhere around the target, and the initial form of the group is not guaranteed to be maintained. For instance, if the concentration value of the robot happens to change from three to two as a result of following other neighbouring robots whose concentration is two, then it starts to follow the

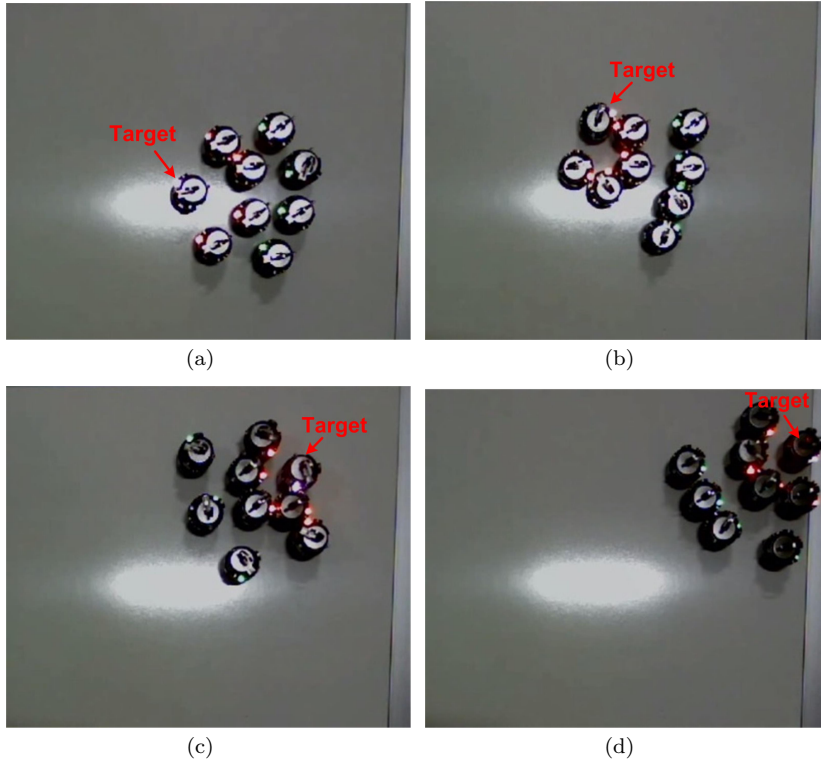
robots whose concentration is one. In this section, to make robots aggregate around a target as well as maintain a desired formation initially given, the difference between the initial concentration value and current value is included in the objective equation along with the attraction term as follows:

$$f = w_a \text{sig}(\Delta d_a, k_a) + w_h |\Delta h| \quad (5)$$

where  $|\Delta h| = h_{ini} - h_{current}$  is the difference between initial and current concentration value, and  $w_h$  is a positive weighting factor. The robots should maintain their initial morphogen concentration, so that they can be positioned at the same relative location from the target as the initial configuration. This is similar to the previous algorithm if the morphogen concentration remains the same during the tracking process; however, if the concentration of the robot is changed, then it follows other robots whose concentration is higher or lower than itself to restore their initial value. Note that this does not guarantee tight formation control of the group, yet can lead to loose maintenance of an initial spatial arrangement during tracking process as shown in Fig. 11. An initial group spatial arrangement remains similar to the initial one in terms of the number of the same coloured robots throughout Fig. 11. Figure 12 shows a preliminary experiment result for an object tracking process with nine kilobots.



**Fig. 11** Object tracking by maintaining the initial morphogen concentration for formation keeping with parameters  $(d_{a,th}, k_a, w_a, w_h) = (69, 0.05, 128, 20)$ .



**Fig. 12** Object tracking by maintaining the initial morphogen concentration with parameters  $(d_{a,th}, k_a, w_a, w_h) = (75, 0.05, 128, 35)$ .

### 3.3 Object Tracking Process Leading to Herding

This section presents an object tracking and herding process. It is realised by combining a light source tracking and a gradient following concept mentioned in the previous sections. For this, along with the objective function in Eq. (5) from gradient following, one more cost on the distance from a light source is added as follow:

$$f = w_a \text{sig}(\Delta d_a, k_a) + w_h |\Delta h| + w_l \text{sig}(\Delta d_l, k_l) \quad (6)$$

where  $w_l$  is a positive weighting factor.  $\Delta d_l = d_l - d_{l,th}$  where  $d_l$  is the distance between the robot and the light source and  $d_{l,th}$  is the threshold distance. Thus, above objective function ensures the robots to maintain their initial morphogen concentration as well as move towards the light source.

Let us assume the situation that one stationary target (at the centre of the swarm group as shown in Fig. 13(a)) is already surrounded by the robots and the part of the group is within the effective range (as solid black circle) of the second target. By applying Eq. (6) to this situation, herding of the first



lack of self-localisation and directional sensing. Several object tracking algorithms were proposed using a light source and morphogen gradient following concept, together with some idea for swarm herding. Numerical simulations and physical experiments showed the effectiveness of the proposed algorithms for tracking and herding of moving targets. Monte Carlo simulations and multiple experiments with different conditions will be performed to analyse the performance statistically as future work. In order to generate more complex and meaningful patterns while efficiently following randomly moving objects or herding them into desired shapes, more sophisticated biological mechanisms such as gene regulatory networks [11] will be exploited. In addition, experiments with larger number of robots will be performed to validate the proposed algorithms.

## References

1. S. Kondo and T. Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329:1616–1620, 2010.
2. C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.
3. A. Okubo. Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Advances in Biophysics*, 22:1–94, 1986.
4. H. Sayama. Robust morphogenesis of robotic swarms. *IEEE Computational Intelligence Magazine*, 5(3):43–49, 2010.
5. M. Mamei, M. Vasirani, and F. Zambonelli. Experiments of morphogenesis in swarms of simple mobile robots. *Applied Artificial Intelligence*, 18:903–919, 2004.
6. R. Nagpal, H. Shrobe, and J. Bachrach. *Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network*. IPSN, LNCS 2634. Springer-Verlag Berlin Heidelberg, 2003.
7. A.F.T. Winfield and J. Nembrini. *Emergent Swarm Morphology Control of Wireless Networked Mobile Robots*, volume 8 of *Morphogenetic Engineering, Understanding Complex Systems*. Springer-Verlag Berlin Heidelberg, 2012.
8. M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3293–3298, 2012.
9. F. Mondadal. The e-puck, a robot designed for education in engineering. In *9th Conference on Autonomous Robot Systems and Competitions*, 2009.
10. Z.Z. Abidin, M.R. Arshad, and U.K. Ngah. An introduction to swarming robotics: application development trends. *Artificial Intelligence Review*, 43(4):501–514, 2015.
11. Y. Jin, H. Guo, and Y. Meng. A hierarchical gene regulatory network for adaptive multirobot pattern formation. *IEEE Trans. on Systems, Man, And Cybernetics-Part B: Cybernetics*, 42(3):805–816, 2012.
12. H. Oh and Y. Jin. Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots. In *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, July 2014.
13. H. Oh, A.R. Shiraz, and Y. Jin. Adaptive swarm robot region coverage using gene regulatory networks. In *Advances in Autonomous Robotics Systems, Lecture Notes in Computer Science*, volume 8717, pages 197–208, 2014.
14. I. Navarro and F. Matia. A survey of collective movement of mobile robots. *International Journal of Advanced Robotic Systems*, 10:1–9, 2013.
15. K. Yeom. Bio-inspired automatic shape formation for swarms of self-reconfigurable modular robots. In *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 469–476, 2010.
16. <https://github.com/ajhalme/kbsim>, 2012.



- 
17. P. Vartholomeos and E. Papadopoulos. Analysis, design and control of a planar micro-robot driven by two centripetal-force actuators. In *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.
  18. I. Navarro and F. Matia. A proposal of a set of metrics for collective movement of robots. In *Proceedings of Workshop on Good Experimental Methodology in Robotics, Robotics Science and Systems*, 2009.