Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
November-December 2018: 1–14
© The Author(s) 2018
DOI: 10.1177/1729881418806435
journals.sagepub.com/home/arx
SAGE

Control algorithms for the emergence of self-organized behaviours in swarms of differential-traction wheeled mobile robots

R Martínez-Clark[1], C Cruz-Hernández[1], J Pliego-Jimenez[1] and A Arellano-Delgado[2]

Abstract

This article proposes three control algorithms for the emergence of self-organized behaviours, including aggregation, flocking and rendezvous, in swarm robotics systems. The proposed control algorithms are based on a local polar coordinates' control law available in the literature for posture regulation; this law is adapted to work in a self-organized robotic swarm using distance and bearing as coupling information. Therefore, the robots only need to know the radial distance and orientation to the goal; additionally, the three algorithms are based on self-organization, eliminating the need for a preset coupling topology among the robots. In particular, the flocking algorithm has a first stage for topology creation, while the rendezvous and aggregation algorithms change the topology on every iteration depending on the local interactions of the robots. The effectiveness of the algorithms was evaluated through numerical simulations of swarms of up to 100 differential traction wheeled mobile robots.

Keywords

Swarm robotics, self-organization, aggregation, flocking, rendezvous

Date received: 14 December 2017; accepted: 15 September 2018

Topic: Mobile Robots and Multi-Robot Systems
Topic Editor: Nak-Young Chong
Associate Editor: M Bernadine Dias

Introduction

Current trends, inspired by the observation of the interaction of social organisms, such as bees, ants and fish, include the implementation of groups to solve tasks in a collective fashion. Many benefits of robotic group implementations have been reported in the literature. The principal advantages of working in groups instead of as individuals for problem solving are summarized below[1]:

- *Ability/capacity*: A single individual may have limited or insufficient capacities to perform tasks alone, especially if tasks are composed of complex subtasks, for example, heavy lifting and simultaneous work in different spaces.
- *Efficiency*: Working in groups allows for tasks to be fulfil faster as well as more effectively and efficiently than by a single individual, particularly if tasks involve constant workspace displacement.

[1] Scientific Research and Advanced Studies Center of Ensenada (CICESE), Ensenada, BC, Mexico
[2] Engineering, Architecture and Design Faculty, Baja California Autonomous University, Baja California, Mexico

Corresponding author:
C Cruz-Hernández, Scientific Research and Advanced Studies Center of Ensenada, Carretera Ensenada-Tijuana No. 3918, Zona Playitas, Ensenada 22860, Mexico.
Email: ccruz@cicese.mx

- *Redundancy and fault tolerance*: Redundancy is the ability of the group to fulfil (totally or partially) the tasks of one group member, assuming the member alone cannot complete the task. This characteristic allows a group to continue working even when some members are lost due to internal failures or by external forces.
- *Cost*: Due to task division, each group member can be constructed with more inexpensive parts, thanks to the limited characteristics needed instead of a single individual performing a complex task alone.

Swarm robotics is an approach of multi-robot systems inspired by the behaviour of social organisms, such as bacteria cultures, ant colonies, bee swarms and bird flocks.[2] The principal objective of this system is to coordinate a group of robots with limited individual characteristics in such way that collective behaviours emerge, to conjointly perform complex tasks.[3] For this purpose, robots depend on the *interaction* with other members of the group in a nearby area, because typically, these robots have short-range communication modules. The characteristics that define a robotic swarm include the following[4]:

- *Robustness*: the ability of the group to fulfil the task collectively in the presence of failures of some swarm members.
- *Scalability*: the capability of the group to complete their tasks indistinctly of the number of members in the swarm.
- *Flexibility*: the adaptation of the robotic swarm to address environment changes.

In the literature, different control strategies are found to generate collective behaviours in multi-robot systems: for example, the emergence of rendezvous using graph theory with fixed coupling among robots is one such strategy.[5,6] On the other hand, it was shown that adaptive control with output feedback produces the same collective behaviour[7]; even more, rendezvous can be achieved with an event-triggered control.[8]

In the flocking case, some works applied graph theory,[9,10] while others were based on a predictive model control scheme.[11,12]

Ultimately, aggregation was treated using commutation techniques for groups consisting of multiple function nodes, imitating the behaviour of certain dopaminergic neurons.[13] In other works, the authors preferred to solve the aggregation problem using probabilistic finite state machines,[14,15] and another paper employed virtual physical forces for the emergence of aggregation.[16] Based on the different literature reported on collective behaviours for multi-robot systems, the objective of this work consists of the present three control algorithms, as well as the numerical simulation results of their implementation on robotic swarms of up to 100 differential traction wheeled mobile robots (WMRs) for the emergence of the following three particular collective behaviours: *aggregation*, *flocking* and *rendezvous*.

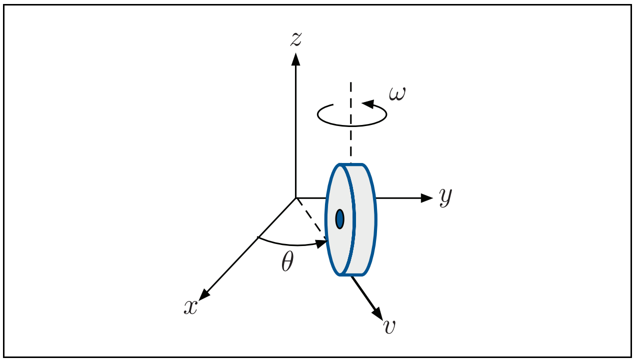


Figure 1. Simplest representation of the unicycle mobile robot.

Nevertheless, all of the collective behaviours reported in this work emerge due to self-organization, a phenomenon commonly found in nature, in which organisms achieve common behaviours without the intervention of external forces or the direct influence of a leader.[17] Some examples of the presence of auto-organization in nature include tissue formation due to the union of distinct cells, pattern formation in desert dunes[18] and labour division in social insect colonies.[19] The main advantages of the control algorithms described in this work are as follows. (i) Unlike evolving techniques, the implemented controllers represent low computational cost. (ii) The inputs needed for the controllers are distance and orientation to other robots or special landmarks on a robot local frame, making them ideal for implementation using ranging sensors and beacons, without the need to use global framework measurement or state observers on the controller. (iii) Unlike in graph-theory-based techniques, a self-organized swarm does not need the robots to have preset IDs nor previous knowledge of the number of robots in the swarm to establish the interactions.

The remainder of the article is organized as follows. In the second section, the definitions of preliminary concepts involved with the kinematics of a single WMR, including the control law for posture regulation for a desired posture input, are presented. In the third section, the proposed control algorithms for the emergence of aggregation, flocking and rendezvous are described. Later, fourth section presents the numerical simulation results of the three control algorithms; finally, in the fifth section, conclusions and future work are stated.

Preliminaries

For the purpose of this work, every robot of the swarm is represented by a *WMR*. The simplest model of a WMR is the unicycle, corresponding to a single upright wheel, rolling in a plane[20] (see Figure 1). With generalized coordinates $\boldsymbol{q} = (q_1, q_2, q_3) = (x, y, \theta)$, the corresponding kinematic model is described by the following equation

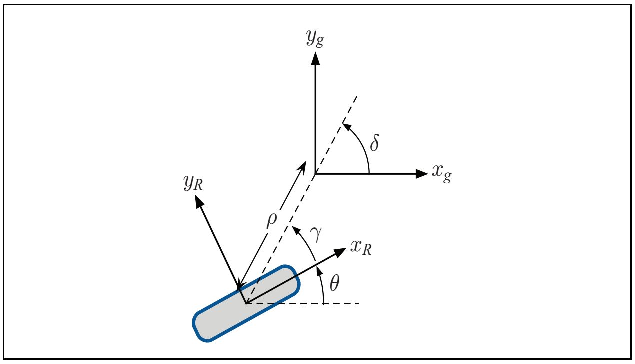


Figure 2. Representation of the unicycle WMR framework in polar coordinates. WMR: wheeled mobile robot.

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{1}$$

in which v and ω represent the linear and angular velocities of the unicycle WMR, respectively.

For the purpose of driving the WMR to a desired position $\boldsymbol{q}_d = (x_d, y_d)$, without loss of generality and considering the desired position as the origin, namely, $\boldsymbol{q}_d = (0, 0)$, we applied the next set of polar coordinates

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ \gamma &= \text{ATAN2}(y, x) - \theta + \pi \\ \delta &= \gamma + \theta \end{aligned} \tag{2}$$

where ρ is the distance of the robot position (x, y) to its desired position (x_d, y_d), γ represents the orientation error angle measured from the robot main axis to the pointing vector from (x, y) to (x_d, y_d) and δ is the orientation with respect to the desired position frame point (see Figure 2). In the new coordinates, the kinematic model (1) becomes

$$\begin{aligned} \dot{\rho} &= v\cos(\gamma) \\ \dot{\gamma} &= v\frac{\sin(\gamma)}{\rho} - \omega \\ \dot{\delta} &= v\frac{\sin(\gamma)}{\rho} \end{aligned} \tag{3}$$

In this work, the following control law in polar coordinates that guarantees asymptotic posture regulation[20,21] is employed

$$\begin{aligned} v &= k_1 \rho \cos\gamma \\ \omega &= k_2\gamma + k_1 \frac{\sin\gamma \cos\gamma}{\gamma}(\gamma - k_3\delta) \end{aligned} \tag{4}$$

with gains k_1, k_2, and $k_3 \in \mathbb{R}^+$.

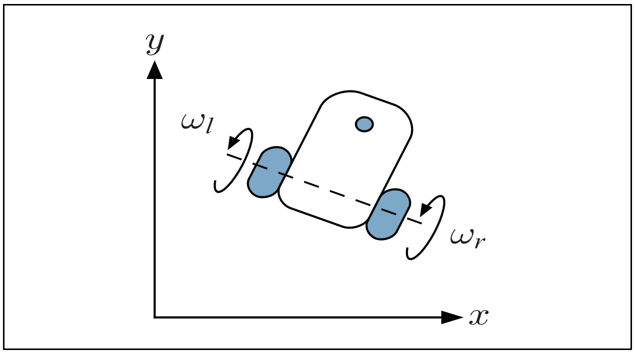


Figure 3. Differential traction WMR. This robot is controlled through the angular velocities of its actuated wheels. WMR: wheeled mobile robot.

The transformation of the controller (4) to polar coordinates entails the existence of a discontinuity, since γ and δ angles are not defined for $\rho = 0$. Due to this restriction, the controller is modified using a switching strategy, similar to the strategy used in the study by Aranda-Bricaire et al.[22] Without loss of generality, the posture regulation of the WMR to the origin of the framework with a desired orientation θ_d is obtained with the next control law

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{cases} \begin{bmatrix} k_1 \rho \cos\gamma \\ k_2\gamma + k_2 \dfrac{\sin\gamma \cos\gamma}{\gamma}(\gamma - k_3\delta) \end{bmatrix} & \text{if } \rho > \varepsilon \\ \begin{bmatrix} 0 \\ k_3(\theta_d - \text{mod}(\theta, 2\pi)) \end{bmatrix} & \text{otherwise} \end{cases} \tag{5}$$

in which $0 < \varepsilon < 1$ corresponds to the regulation precision threshold, and mod represents the module operation.

For WMR implementation purposes, a *differential traction* structure is employed, which corresponds to the kinematics of a unicycle robot with simplest construction and control. This WMR model is controlled through the modulation of the angular velocities of its actuated wheels (see Figure 3). If the control law (4) is used to regulate the posture of a WMR with differential traction, it is necessary to calculate the angular velocities of its left and right wheels, ω_l and ω_r, respectively. This calculation can be achieved from the linear and angular velocities of the robot, the radius r of its wheels and the distance d between them using the following transformation

$$\omega_l = \frac{v - \left(\frac{d}{2}\right)\omega}{r}, \qquad \omega_r = \frac{v + \left(\frac{d}{2}\right)\omega}{r} \tag{6}$$

Self-organized collective behaviours

This section describes the following three collective behaviours generated in this work: aggregation, flocking and rendezvous, as well as the corresponding control strategies.

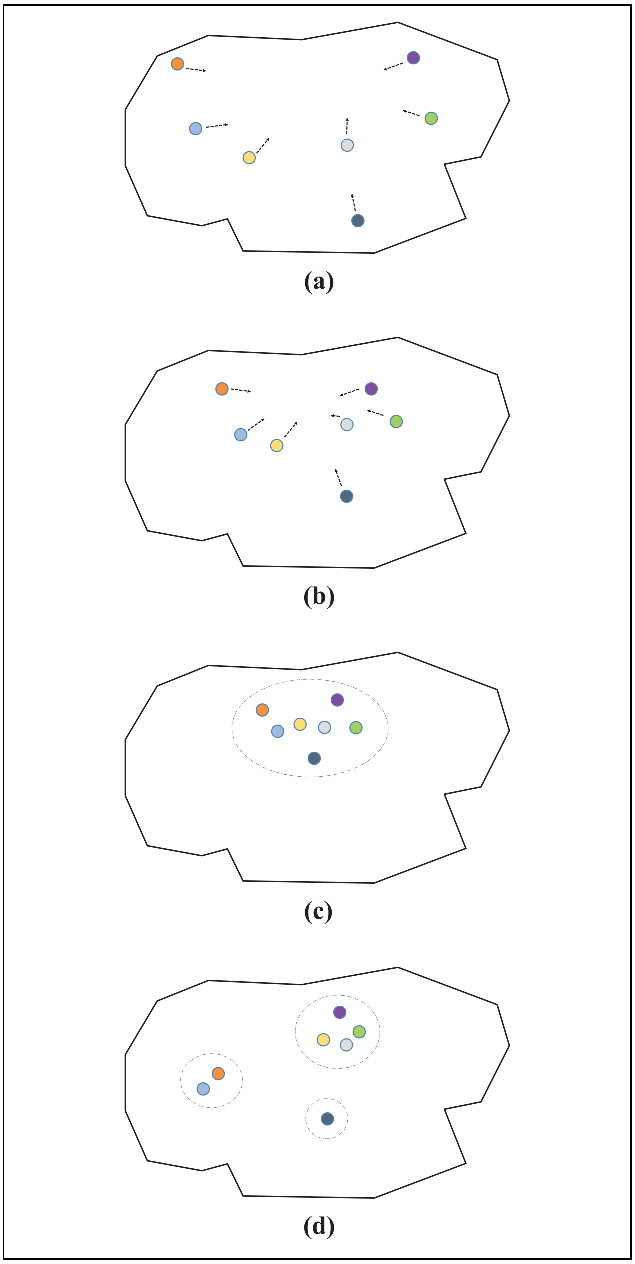


Figure 4. Aggregation of a robotic swarm: (a) first, all the robots are scattered; (b) next a strategy is implemented to select a zone to gather; (c) depending on the strategy, a single group can be formed or (d) may form several groups, including isolated robots.

Aggregation

Aggregation is a collective behaviour frequently observed in natural organisms, ranging from bacteria to social insects, such as bees, termites and cockroaches as well as in various mammals.[23] Aggregation helps organisms avoid predators, endure hostile environments and reproduce.[24] This behaviour emerges in two different ways. First, the behaviour can be guided by the ambient conditions; for instance, organisms gather in environment zones with propitious characteristics such as temperature, humidity or resource availability. Second, this behaviour can emerge in homogeneous environments without special zones; in this case, the aggregation is a product solely of the interactions among the members of the group.[25] Concerning robotic swarms, the objective of aggregation (Figure 4) is to group an originally scattered set of robots in a particular environment zone.[4] This basic behaviour is fundamental because it allows the robots to get close in such a way that more interactions exist, producing more complex collective behaviours.[26] The principal approaches for the emergence of this behaviour reported in the literature include artificial evolution, probabilistic algorithms and artificial potential fields.[27]

In this work, the last approach is used to achieve aggregation in a robotic swarm. The strategy consists of driving every robot to the centroid of a triangle formed by its position, the position of the closest robot to it and the robot corresponding to the farthest robot in its sensing area (see Figure 5). To accomplish this task, every robot behaves according to Algorithm 1.

Algorithm 1: Self-organized aggregation procedure

```
procedure FOR THE $i$–TH ROBOT:
loop:
        Calculate distance to all robots in range
    $d_{prox} \leftarrow$ distance to nearest robot
    $d_{far} \leftarrow$ distance to farthest robot
        $x_d = \dfrac{x_i + x_{prox} + x_{far}}{3}$
        $y_d = \dfrac{y_i + y_{prox} + y_{far}}{3}$
        calculate $\omega_l$ and $\omega_r$ with equations (5) and (6)
go to loop.
```

It is important to establish that the efficiency of this strategy resides in the communication radius of each robot, which, if too small, can induce the formation of different scattered groups and even isolate some robots from the groups. To avoid this issue, some algorithms use external signals, which allow the robots to detect special aggregation areas.[28]

Flocking

Flocking is often present in nature, for example, in birds, fish and animals flock. Decades ago, scientists of different disciplines, including animal behaviour, physics, social sciences and computational sciences, have dedicated themselves to study the emergence of this collective behaviour.[29] This behaviour is characterized by a *fluent movement of the whole group*, due to individual actions of every member of the group (Figure 6). Reynolds[30] proposed a set of basic rules for local interaction between elements of the group with the purpose of coordinating their movements, eventually known as the Reynolds rules[31]:

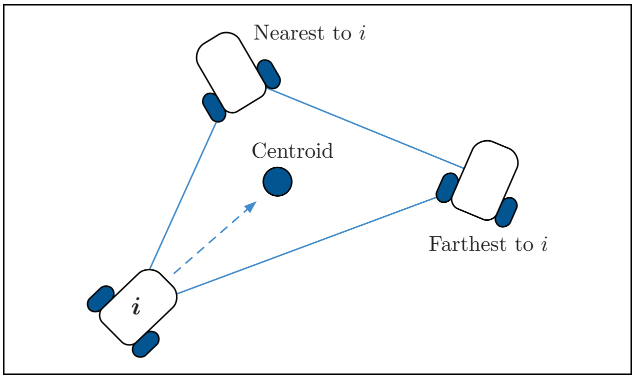


Figure 5. The *i*th robot tries to reach the position of the centroid in the figure formed between its position and those of the nearest and farthest robots to it.

- *Cohesion*: the elements of the swarm should stay together.
- *Alignment*: the elements should match their speeds.
- *Separation*: the elements in the swarm should avoid collision.

As an application example, flocking was achieved, combining the Reynolds rules with a spiral search algorithm, based on the behaviour of hunting hawks, for a swarm of autonomous underwater vehicles with searching purposes.[32] On the other hand, Tanner et al. studied the stability in a swarm formed by agents with double integrator dynamics, in which every agent is controlled in a way that all the agents move with a common velocity while maintaining a specific distance among them using two different approaches: fixed coupling topologies[33] and time-varying topologies.[34]

To generate this collective behaviour in a robotic swarm, we propose that WMRs move like a platoon, displaying a chain pattern while following a specific trajectory. To achieve flocking, the concept of n-trailer is used, in which a robot drags an arbitrary number of trailers represented with more WMRs.[35] In this case, we identify the following two different control objectives: one for the first robot, which follows the desired trajectory, and one for the follower robots, which try to regulate their positions to an anchor point located at a distance ℓ behind another robot (see Figure 7). This point represents the anchorage between the truck and the trailer.

For the first robot in the platoon, the posture regulation controller (5) is used, in which, the desired position $\boldsymbol{q}_d(t) = \left(x_d(t), y_d(t)\right)$ is calculated as a series of desired points in the plane in such a way that they represent a specific trajectory. Therefore, the desired position of this WMR is calculated with a time-varying function of this type

$$x_{d1}(t) = f(x, y, t), \quad y_{d1}(t) = f(x, y, t) \tag{7}$$

The path generator must satisfy the non-holonomic constraint in the WMR. For the follower robots, the posture regulation controller (5) is used to drive the robots to their anchor point. In this case, the desired position for the ith WMR applied at the input of its controller is selected as follows

$$\begin{aligned} x_{di} &= x_{i-1} + \ell \sin\left(\theta_{i-1} - \frac{\pi}{2}\right) \\ y_{di} &= y_{i-1} - \ell \cos\left(\theta_{i-1} - \frac{\pi}{2}\right) \end{aligned} \tag{8}$$

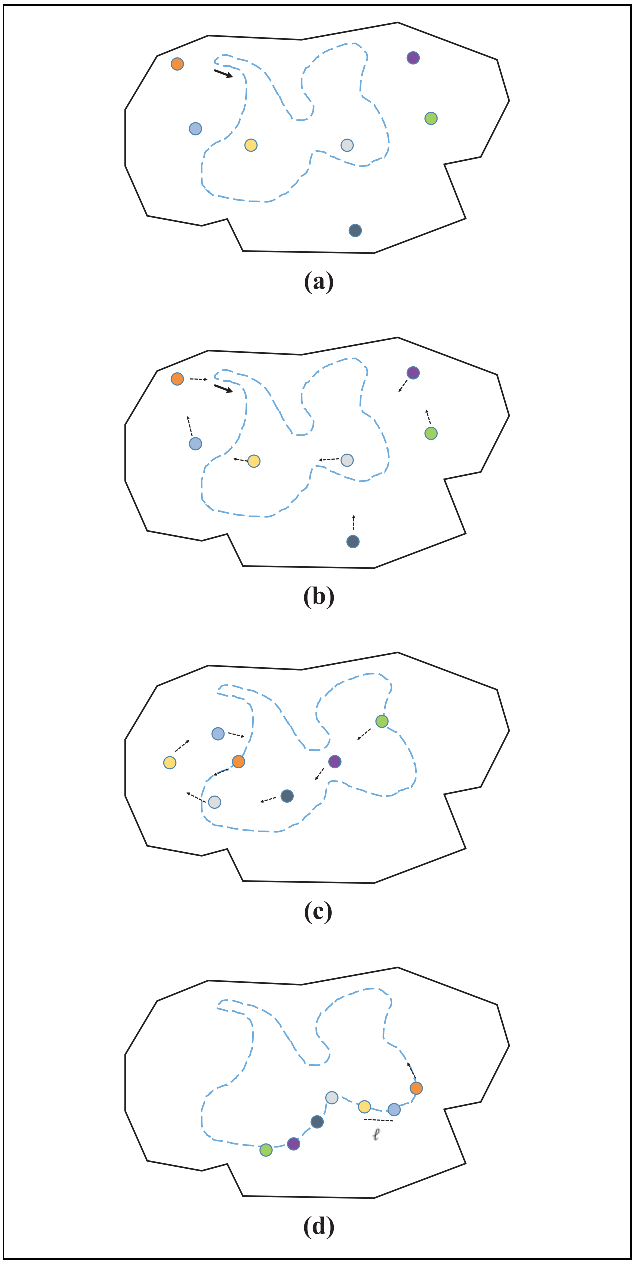


Figure 6. n-Trailer flocking. (a) Robots need to order themselves to display the desired formation. (b) The robot is designated with $id = 1$ and tries to follow the desired trajectory while the rest of the robots tries to align behind the other robot. (c) Robot 1 follows the trajectory, while the other robots continue aligning themselves until (d) each robot is at a distance ℓ to another robot.

It is worth to mentioning that, for the self-organization of the swarm, all robots have both behaviours programmed, and it is up to the group to select the robot that will follow the desired trajectory. For this purpose, every WMR executes Algorithm 2, based on equations (7) and (8).

Algorithm 2: Self-organized flocking

```
procedure FOR EACH ROBOT:
    Calculate distance to starting point of trajectory
    Compare dist. to the start of traj. with other robots
  $robot_{id}$ ← sort robots from nearest the start of traj.
  loop:
    if $robot_{id} = 1$ then
        $x_d(t)$ ← depends on trajectory
        $y_d(t)$ ← depends on trajectory
    else
        $x_{d_{id}}(t) = x_{d_{id-1}}(t) + \ell \sin(\theta_{id-1} - \pi)$
        $y_{d_{id}}(t) = y_{d_{id-1}}(t) + \ell \cos(\theta_{id-1} - \pi)$
    calculate $\omega_l$ and $\omega_r$ with equations (5) and (6)
    go to loop.
```

Rendezvous

In a robotic swarm acting in a finite environment, there may exist some in which these robots must gather in a determined spot; this behaviour is known as *rendezvous* (Figure 8). This collective behaviour can be the first step of pattern formation, or this spot can be used as a recharging or maintenance station, also, it can be used to start complex behaviours such as collective transportation .[36] In the literature, different definitions of rendezvous are reported; however, in this work, it is considered as the task to *simultaneously* gather a group of robots in a previously known (or negotiated) place in the workspace.[37,38] If t_{si} represents the settling time of the ith robot, a swarm of N robots achieves rendezvous if the following is true

$$t_{s1} = t_{s2} = \ldots = t_{sN} = t_{\text{rdvz}} \quad \text{if } (x_{d_i}, y_{d_i}) = (x_{\text{rdvz}}, y_{\text{rdvz}}), \quad \text{with } i = 1, 2, .., N \tag{9}$$

It is possible to apply different control strategies to achieve this objective. For example, in the literature, works show that rendezvous is achieved by applying a series of advance and stop movements on the robots until they gather.[39,40] In this work, the proposed strategy consists of amplifying and attenuating the linear and angular velocities of the regulation controller (5) with variable gain, which is a function of the distance from the robots to the rendezvous point. Thus, we assume that the WMR moves in a finite workspace in which the distance from any robot to the rendezvous point is bounded by $\rho_{\max}$. Taking as an objective the simultaneous arrival of robots i and j to a rendezvous point, the variable gain K_{rdvz_i} of robot i is obtained as

$$K_{\text{rdvz}_i} = 1 + \left(\frac{\rho_{r_i} - \rho_{r_j}}{\max(\rho_{r_i}, \rho_{r_j})} \right) \tag{10}$$

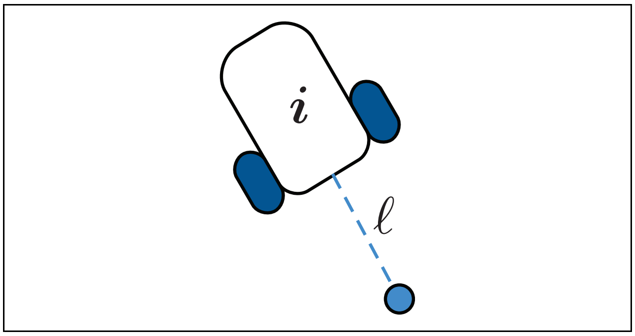


Figure 7. Coupling point of the $i + 1$ robot, located at a distance ℓ behind robot i.

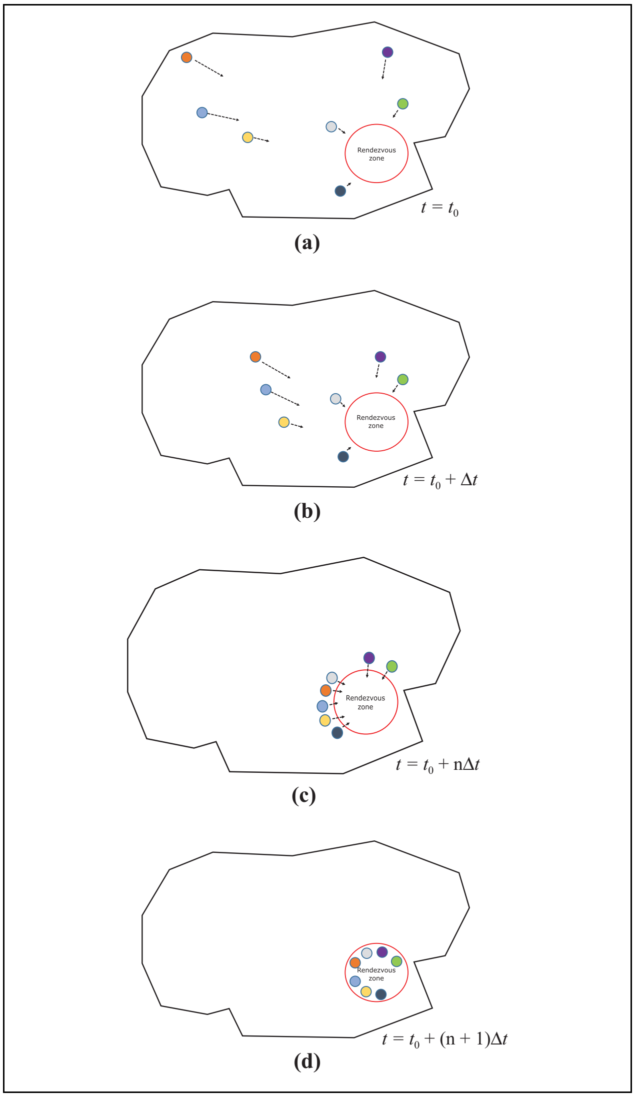


Figure 8. Rendezvous of a robotic swarm. (a) A group of scattered robots try to regulate their position to a previously known rendezvous zone. (b) The farthest robots come closer to the zone faster, while robots closest to the rendezvous zone decelerate, waiting for the others. (c) In an arbitrary time step, all robots lie outside the border of the rendezvous zone, while in the next time step, (d) all robots cross the rendezvous zone border simultaneously.

in which ρ_{r_i} and ρ_{r_j} represent the distance of robots i and j to the rendezvous point, correspondingly. In this manner, the controller (5) is modified as follows

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{cases} K_{\mathrm{rdvz}_i} \begin{bmatrix} k_1 \rho_i \cos\gamma_i \\ k_2\gamma_i + k_2 \dfrac{\sin\gamma_i \cos\gamma_i}{\gamma_i}(\gamma_i - k_3\delta_i) \end{bmatrix} & \text{if } \rho_i > \varepsilon \\ \begin{bmatrix} 0 \\ k_3(\theta_{d_i} - mod(\theta_i, 2\pi)) \end{bmatrix} & \text{otherwise} \end{cases} \tag{11}$$

It can be observed in equation (10) that if robot i has a greater distance to the rendezvous point than robot j, that is $\rho_{r_i} > \rho_{r_j}$, the velocities of robot i are amplified because $1 < K_{\mathrm{rdvz}_i} < 2$. On the other hand, if $\rho_{r_i} < \rho_{r_j}$, the velocities of robot i are attenuated with a gain $0 < K_{\mathrm{rdvz}_i} < 1$. With the purpose of achieve rendezvous in a swarm with more than two WMRs, we propose to use a controller (11). Therefore, for each time step, every robot computes the distance to all robots in its sensing area and its gain (10) with the closest robot stored as d_{prox} if and only if the average distance to the rendezvous point of the robot and the closest robot to it is greater than a threshold distance $d_{\mathrm{lim}} > 0$. Otherwise, the robot computes its gain (10) with the farthest robot to it, labelled as d_{far}. This threshold distance was included to consider that the robots near to the rendezvous zone have to wait for robots that are far from the rendezvous zone. Thus, because of self-organization in every iteration of the control algorithm, the network topology can change because of the different robots selected for gain calculation. This behaviour considers that as many robots as possible will rendezvous together. Therefore, each robot in the swarm behaves according to Algorithm 3.

Algorithm 3: Self-organized rendezvous

```
procedure FOR $i$—TH ROBOT:
    loop:
        Calculate distance to all robots in range
        $d_{prox} \leftarrow$ nearest robot
        $\rho_{r_i} \leftarrow$ dist. between robot i and rendezvous point
        $\rho_{r_{iprox}} \leftarrow$ dist. between $d_{prox}$ and rndz. point
        if $\frac{\rho_{r_i}+\rho_{r_{iprox}}}{2} < d_{lim}$ then
            $d_{far} \leftarrow$ farthest robot
            $\rho_{r_{iprox}} \leftarrow$ dist. between $d_{far}$ and rndz. point
        $K_{rdvz_i} = 1 + \frac{\rho_{r_i} - \rho_{r_{iprox}}}{\max(\rho_{r_i}, \rho_{r_{iprox}})}$
        calculate $\omega_l$ and $\omega_r$ with equations (6) and (11)
        go to loop.
```

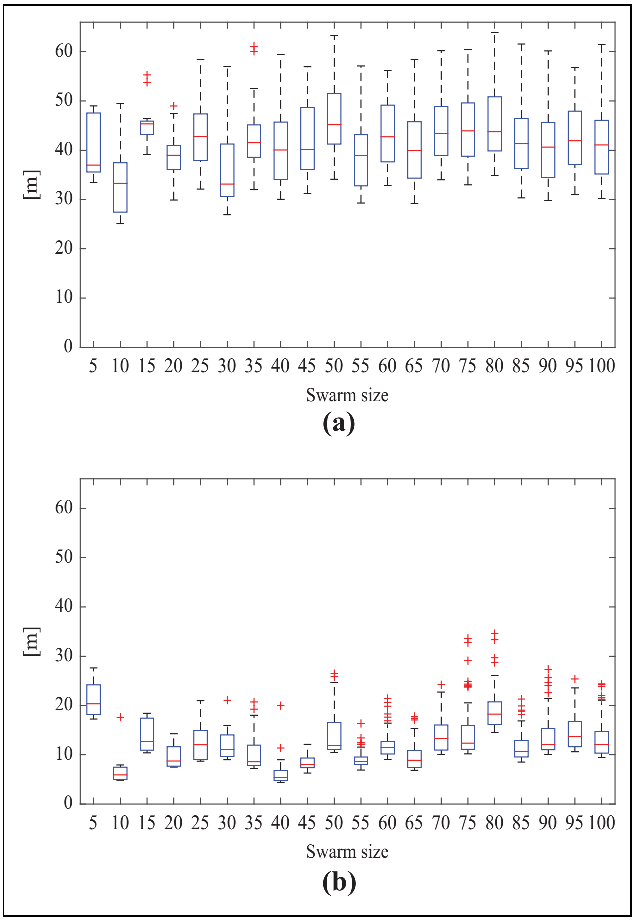


Figure 9. Results of aggregation. (a) Initial average distances of the robots with a box plot. (b) Average distances of the robots after 2000 aggregation algorithm iterations in all simulated swarms.

Numerical results

To assess the effectiveness of the proposed control algorithms, numerical simulations on swarms of different sizes were conducted to evaluate the scalability. For all cases shown, the initial postures $\boldsymbol{q}_i(0)$, $i = 1, \ldots N$ of the robots were taken randomly within an 80×80 m^2 workspace *without obstacles*. For aggregation and rendezvous, the results shown for the swarm simulations with 5, 10, 15, ..., 100 WMRs and for flocking are the results of swarms containing 5 and 10 WMRs. In all simulations, the WMRs have a limit velocity on their wheels of 50 rad/s and are represented by robots with a wheel radius of 20.5 mm and a distance between wheels of 88.41 mm.

Aggregation

In this case, the algorithm was iterated 2000 times for each robotic swarm with sampling period $t_s = 0.22$ s. According to Algorithm 1, each robot calculates its distance with the other robots in its sensing range. In Figure 9(a), a comparison of the initial average distances for each robotic swarm is shown, while Figure 9(b) shows how the average distance between robots diminished after execution of the algorithm.

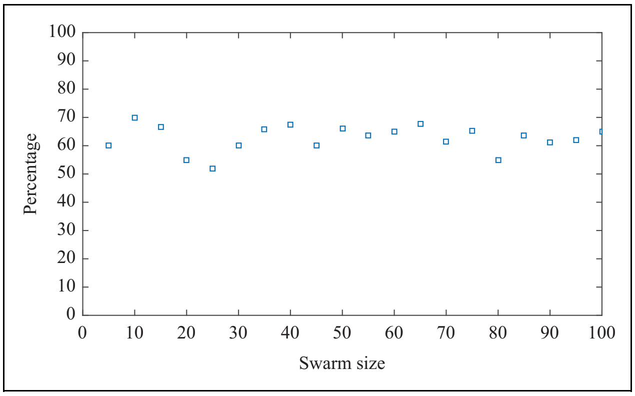


Figure 10. Percentage of robots belonging to the main aggregate for different simulated swarms.

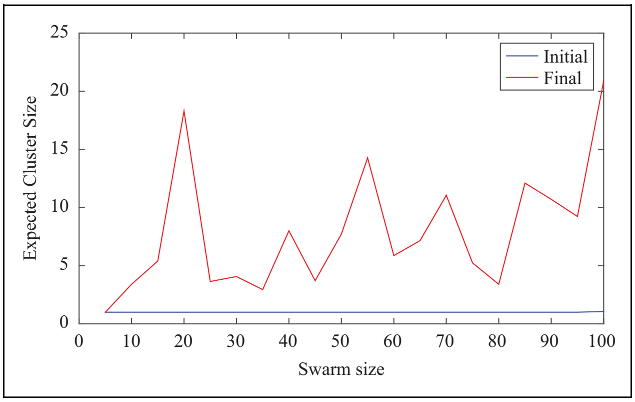


Figure 11. ECS for different swarm sizes. ECS: expected cluster size.

Due to the nature of the algorithm, some robots can be isolated from the main aggregate (i.e. the group with more robots), forming another aggregate with fewer elements. It is difficult to delimit the area covered by the main aggregate, thus the use of the average distances between all robots is proposed as a threshold for belonging to main aggregate. Therefore, a robot belongs to the main aggregate if its average distance to the other WMRs is equal to or less than the average distances of all robots. Figure 10 shows the percentage of WMRs that belong to the main aggregate considering this metric. In the literature, different metrics are used to evaluate aggregation algorithm performance. In particular, the *expected cluster size* (ECS) and *total distance* (TD) among robots are examiend.[14] ECS uses a threshold T_{close} to determine robots in the same cluster. Robots closer than T_{close} are considered neighbours. if dist (R_i, R_j) denotes the distance of ith to jth WMR, neighbourhood relationship is defined as follows[14]:

$$\text{Neigh}(R_i, R_j) = \begin{cases} 1 & \text{if dist}(R_i, R_j) < T_{\text{close}} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Using this neighbourhood, the size of a cluster can be defined for each robot; namely, $\text{size}(R_i)$ denotes the number of robots that belongs to the ith robot cluster. With this cluster size, the average of cluster sizes can be computed for each robot in an n-robot swarm as follows

$$\text{ECS} = \frac{1}{n}\sum_{i=1}^{n} \text{size}^2(R_i) \tag{13}$$

The other metric included in this work is TD, which measures the TD between each robot pair. This metric uses negative distance to emphasize the high metric value for better clustering. TD is defined as follows[14]

$$\text{TD} = -\sum_{i=1}^{n}\sum_{j=i+1}^{n} \text{dist}(R_i, R_j) \tag{14}$$

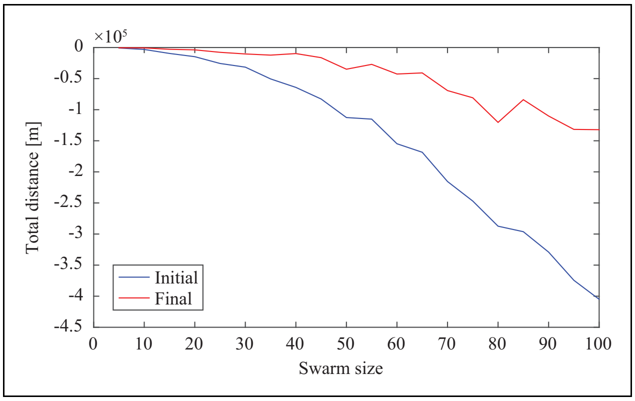


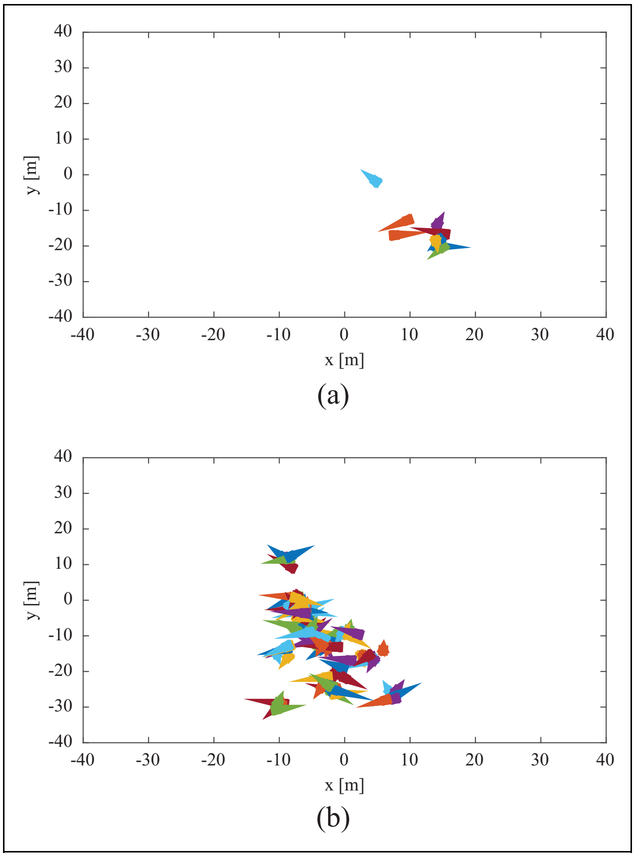
Figure 12. TD among robots before (blue) and after (red) execution of the aggregation algorithm. TD: total distance.

We can observe the estimated cluster size for each simulated swarm size, considering $T_{\text{close}} = 1$ m, as shown in Figure 11. At the beginning of the simulation, all robots are scattered through the workspace, and the cluster size is equal to one for each robot. After algorithm execution, the cluster sizes are observed to grow in each case.

On the other hand, regarding the spatial information, Figure 12 shows the TD among robots for each swarm size. It is worth mentioning that distance decreases with 2000 algorithm iterations.

Figure 13(a) shows the aggregation of a 10-WMR swarm; in this case, the main aggregate contains 7 WMRs, while 3 WMRs are scattered through the workspace. Two scattered WMRs aggregate themselves close to the main aggregate. On the other hand, Figure 13(b) shows that some secondary aggregates formed around the main aggregate in the 100-WMR swarm.

As an example, Figure 14 shows the average distance of each WMR in the 10-WMR swarm of Figure 13(a), highlighting the average distances metric used to determine whether a WMR belongs to the main aggregate.



(a)

(b)

Figure 13. Positions of the robots in a swarm of (a) 10 robots and (b) 100 robots after execution of the aggregation algorithm.

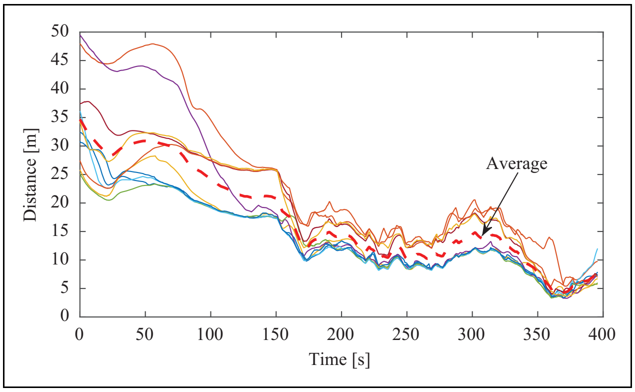
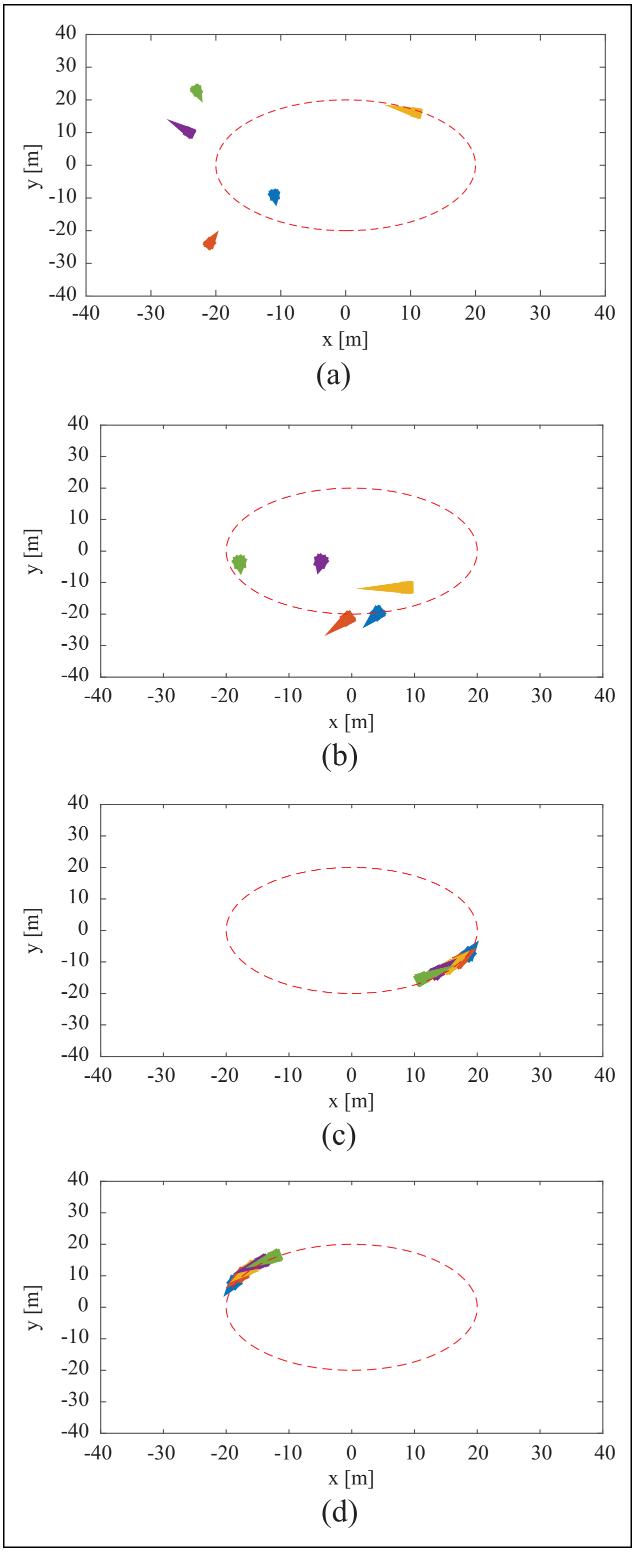


Figure 14. Average distances of 10 robots in the swarm over time. The dashed line represents the average distances.

Flocking

In flocking simulations, Algorithm 2 was iterated 2000 times with a sampling period of 0.22 s. The desired distance between the WMRs was selected as 0.5 m. The desired trajectories for the simulations are circular, and a Gerono's lemniscate with frequency of $2\pi/200$ rad and radius of 20 m was used for 5- and 10-WMR swarms.

For the first simulation, Figure 15 shows the postures of five WMRs at the beginning of the simulation and for different time instants. The WMRs describe a circular trajectory while maintaining a chain formation. Figure 16 shows the WMR postures at different time instants while



(a)

(b)

(c)

(d)

Figure 15. Different time instants for the flocking simulation of a five-robot swarm tracking a circular trajectory. Postures correspond to (a) initial, (b) $t = 9.65$ s, (c) $t = 39.56$ s and (d) $t = 139$ s.

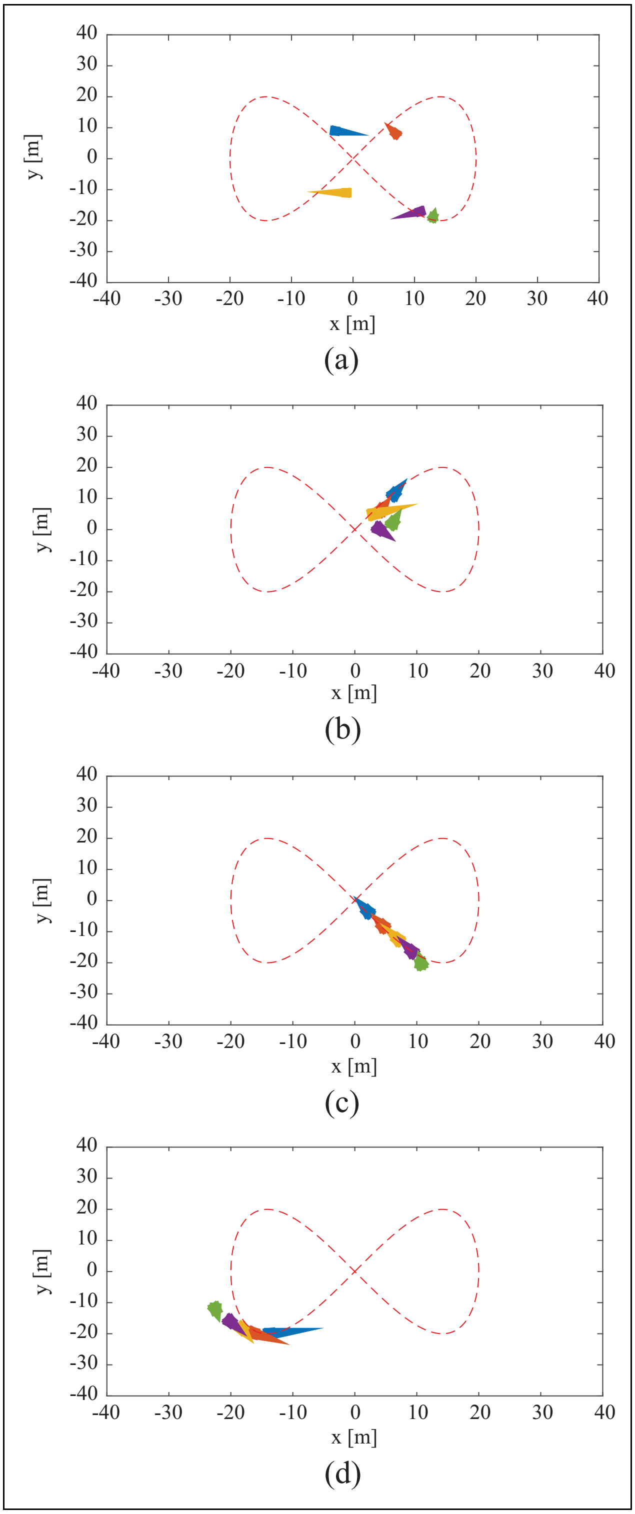


Figure 16. Postures of the five robots in the swarm tracking Gerono's lemniscate. Postures correspond to (a) $t = 0$ s, (b) $t = 11.84$ s, (c) $t = 98.95$ s and (d) $t = 177.73$ s.

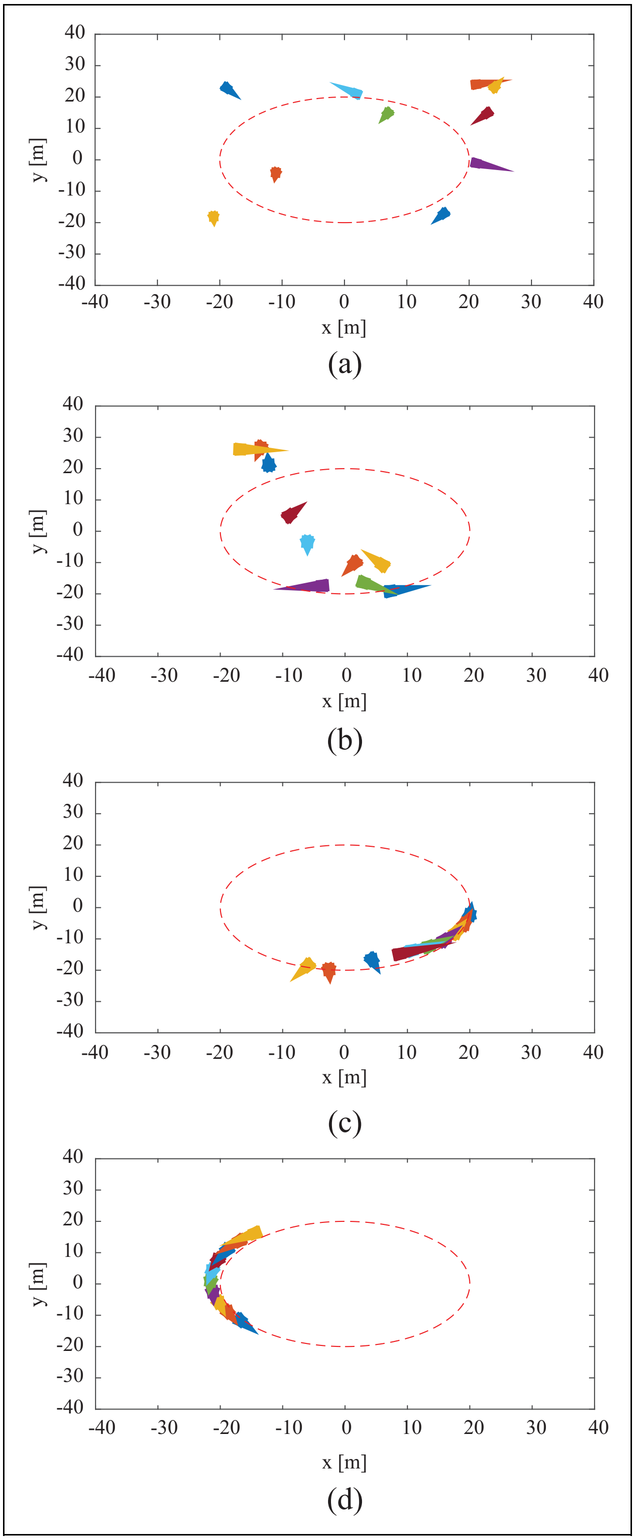


Figure 17. Results of flocking in a 10-robot swarm in a circular trajectory. Postures correspond to (a) initial, (b) $t = 14.50$ s, (c) $t = 48.56$ s and (d) $t = 171.50$ s.

tracking Gerono's lemniscate trajectory. In both cases, the red dashed line corresponds to the desired trajectory.

For the simulations corresponding to the 10-WMR swarm, Figure 17 shows the behaviour of the WMRs at different time instants while trying to flock in a circular trajectory. Figure 18 shows the corresponding postures for the same swarm pursuing Gerono's lemniscate

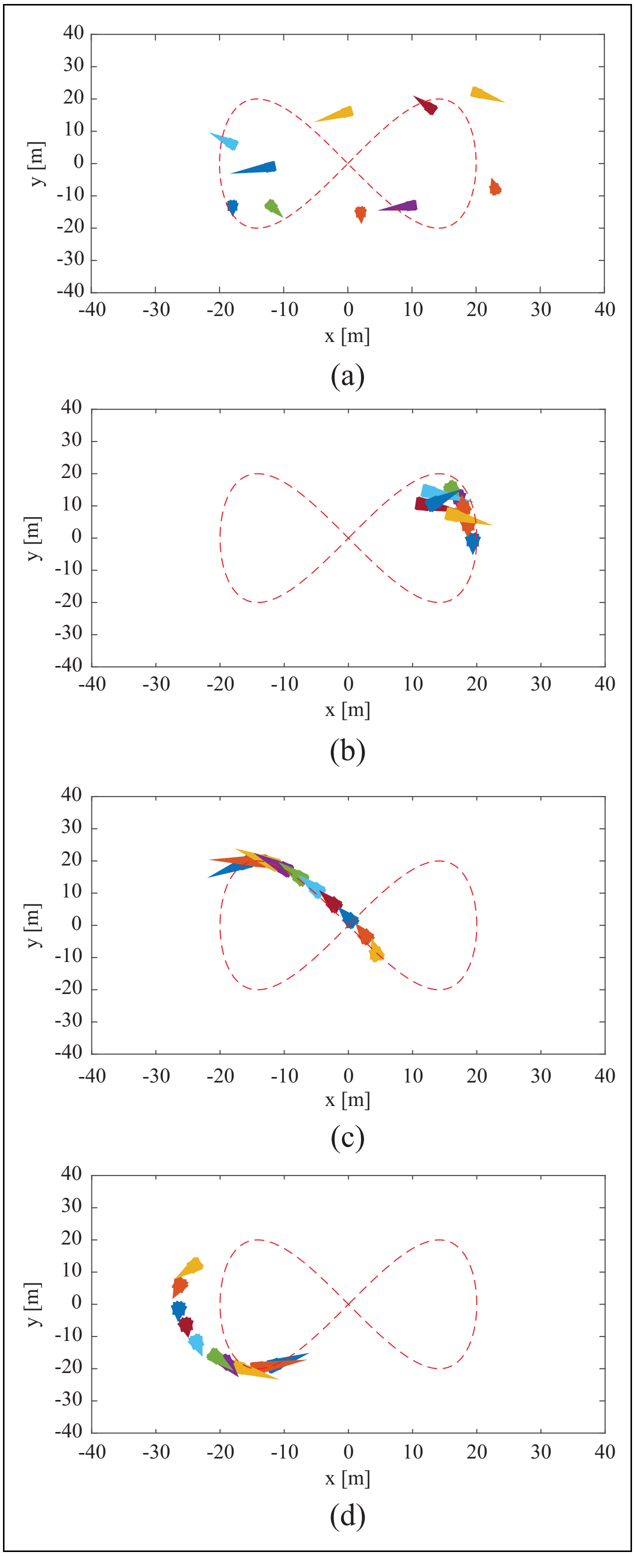


Figure 18. Swarm of 10 robots describing Gerono's lemniscate while moving like a platoon. Postures correspond to (a) $t = 0$ s, (b) $t = 65.54$ s, (c) $t = 163.82.45$ s and (d) $t = 227.45$ s.

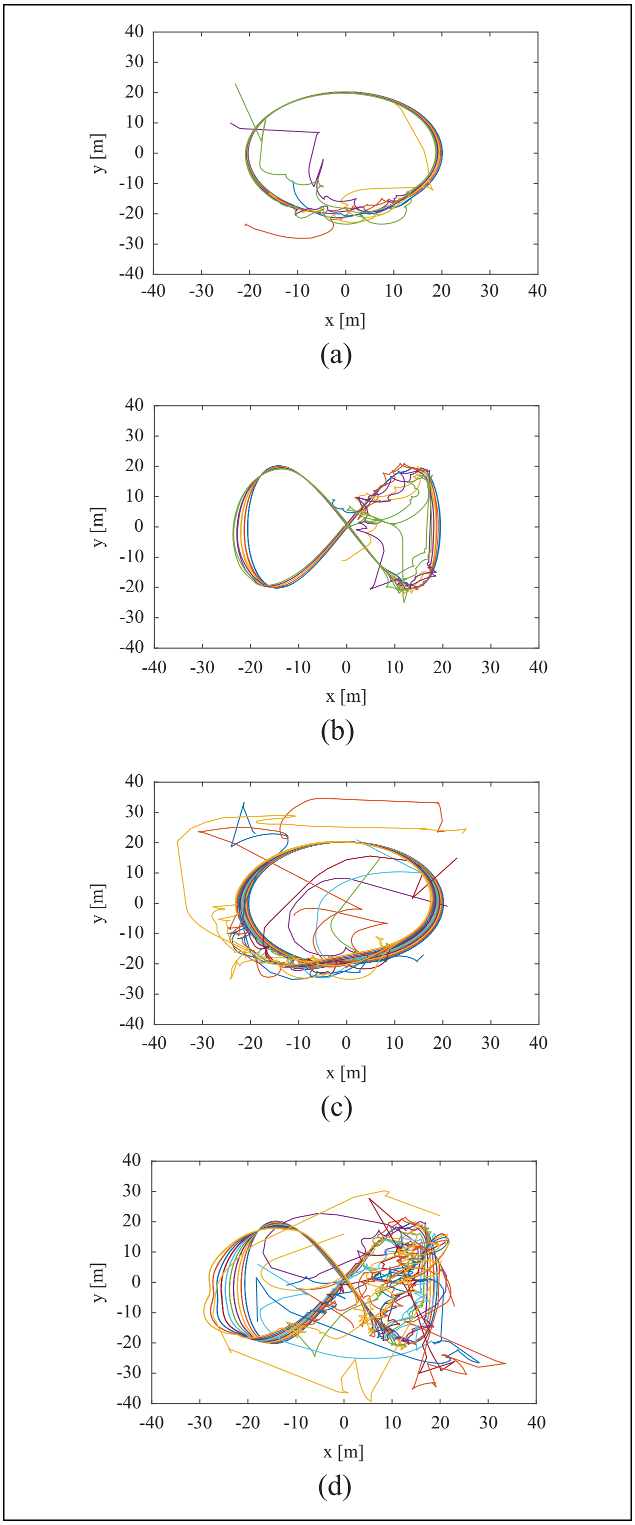


Figure 19. Trajectories described for the robots flocking in the four simulations. (a) Five-robot swarm with circular trajectory, (b) Gerono's lemniscate trajectory and the 10-robot swarm, (c) circular and (d) Gerono's lemniscate.

trajectory. As in previous figures, the desired trajectory of the first WMR in the platoon is represented with the red dashed line.

Figure 19 shows the trajectories of the robots in the different simulations. As the number of robots increases, it takes more time to make the selected formation because

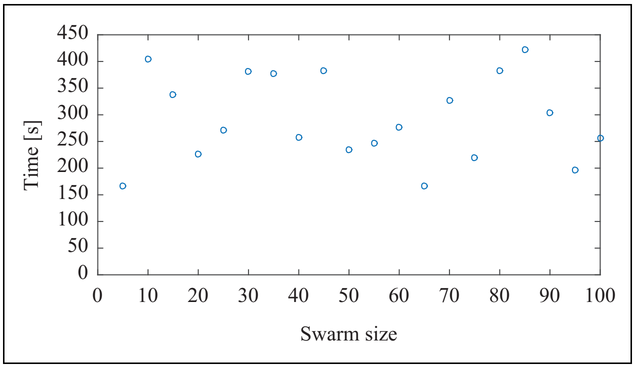


Figure 20. Rendezvous time for different swarm sizes.

the platoon formation forces some robots to wait and do extra movements to solve the non-holonomic constraints inherent to the unicycle WMR.

Rendezvous

For this case, the rendezvous algorithm was iterated 2000 times for each robotic swarm. It is worth highlighting that the rendezvous point changed to a rendezvous zone, in which the robots are considered gathered, with a radius of 5 m centred on the desired rendezvous point. For all cases, this point is previously known for the robots and was selected randomly within the workspace. Figure 20 shows the time needed to rendezvous all the robots, meaning all the robots are in the rendezvous zone.

As an example, Figure 21 shows the behaviour of the WMRs in a 10-WMR swarm when they are about to rendezvous around 403.9 s, corresponding to 1836 iterations of the rendezvous algorithm with a sampling period of 0.22 s.

The initial postures for the 100-WMR swarm are shown in Figure 22. The time needed to rendezvous this swarm was 256.3 s, as seen in Figure 20 and corroborated with the position errors of Figure 23. Notice how the distance errors e_r of all the WMRs simultaneously cross the red dashed line corresponding to the radius of the rendezvous zone.

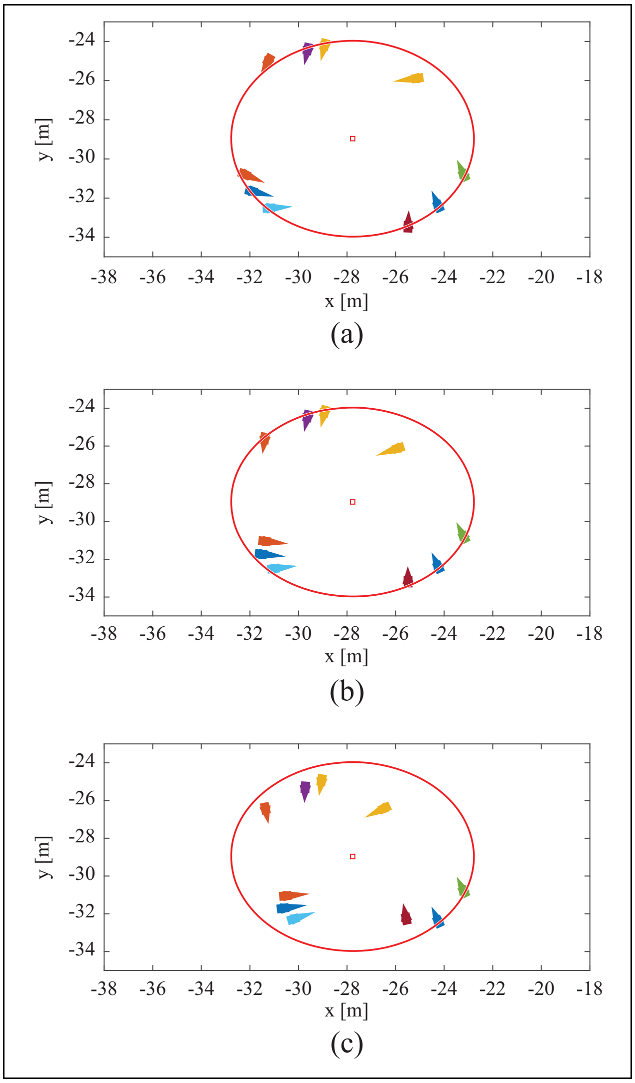


Figure 21. Postures of the robots around the rendezvous zone (delimited with the red circle) at (a) $t = 402.38$ s, (b) $t = 403.70$ s and (c) $t = 405.02$ s.

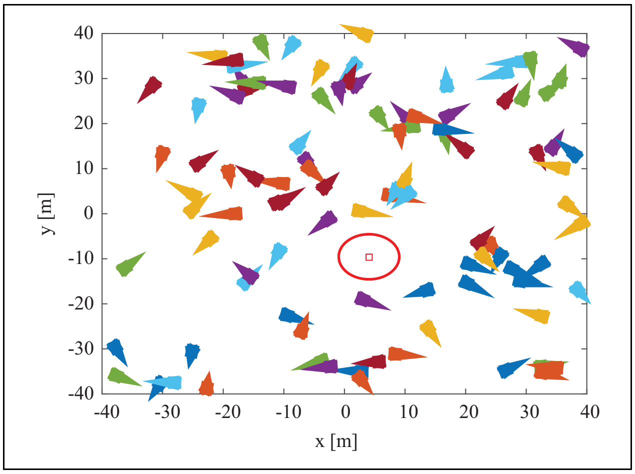


Figure 22. Initial postures of 100 robots in the swarm for the rendezvous simulation.

Conclusions

In this work, using a posture regulation controller found in the literature, three control algorithms were developed for the emergence of self-organized collective behaviours in a robotic swarm. The proposed algorithms are based on polar coordinates; therefore, the robots share only distance and bearing to special neighbouring landmarks. None of the algorithms need a preset static coupling topology due to self-organization strategies. Future work includes implementation on real robotic swarms and examining the implementation feasibility in the design process of the algorithms. The algorithm simplicity in all the cases is found in the use of distances and orientations to other robots or special landmarks; therefore, implementation is intended to be achieved using ranging sensors and beacons.

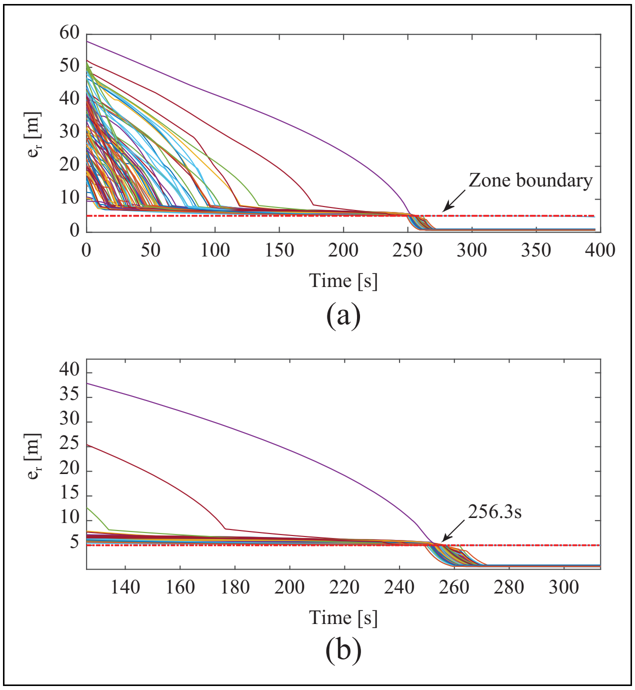


Figure 23. (a) Position error of robots with respect to the rendezvous point and (b) zoom of the errors around the rendezvous time. For both graphs, we can observe the distance to the rendezvous point, highlighting the rendezvous area radius with the dashed line.

None of the proposed algorithms need position calculation in the global framework. The control algorithms were evaluated through numerical simulations on swarms up to 100 WMRs, selected for illustration purposes, in order to prove scalability. Future work will find a bound on the number of robots that can be controlled at once. Additionally, none of the proposed algorithms consider collision avoidance; thus future work will implement a strategy to avoid collisions between robots. In the aggregation algorithms, the results show it is not feasible in every case to aggregate all the robots in a single group; although, robots outside the group considerably decreased their average distance to other robots. In some cases, more than one group emerged. The implemented flocking strategy is not unique, because different formations can be obtained by changing the position of the anchor point. In that sense, the formation pattern can be changed to emulate the pattern displayed by migrating flocks or to represent schools of fish. From the numeric results, we conclude that the size of the swarm and the complexity of the trajectory are fundamental for the position allocation of robots because a contradiction may arise from the platoon length and the desired trajectory. In the rendezvous task, we considered that all the robots have to arrive to the rendezvous point simultaneously; however, it is more practical to consider the idea of a rendezvous area in which all robots are gathered. For all swarm sizes simulated, a practically simultaneous arrival of robots was achieved at the border of the rendezvous zone.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by CONACYT through the Research Project on Basic Science, ref. 166654 (A1-S-31628).

ORCID iD

C Cruz-Hernández https://orcid.org/0000-0002-5915-6806

References

1. Anderson C and Franks NR. Teamwork in animals, robots, and humans. In: Slater PJB, Rosenblatt JS, Snowdon CT, and Roper TJ (eds.); Naguib M (Collaborator), *Advances in the study of behavior*, Vol. 33. San Diego: Elsevier Academic Press, pp. 1–48.
2. Tan Y and Zheng ZY. Research advance in swarm robotics. *Defence Technol* 2013; 9(1): 18–39.
3. Sahin E. *Swarm robotics*, *lecture notes in computer science*, Vol. 3342. Berlin: Springer Berlin Heidelberg, 2005.
4. Brambilla M, Ferrante E, Birattari M, et al. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell* 2013; 7(1): 1–41.
5. Kingston D, Ren W, and Beard R. Consensus algorithms are input-to-state stable. In: *Proceedings of the american control conference*, Portland, OR, USA, 8–10 June 2005, pp. 1686–1690. IEEE.
6. Ren W, Beard RW, and McLain TW. Coordination variables and consensus building in multiple vehicle systems. In: Kumar V, Leonard N, and Morse AS (eds) *Lecture notes in control and information sciences*, vol. 309, 2004, pp. 171–188. Berlin, Heidelberg: Springer.
7. Li P, Xu S, Ma Q, et al. Leader-following rendezvous for uncertain Euler–Lagrange multi-agent systems by output feedback. *J Frankl Inst* 2017; 354(10): 4215–4230.
8. Zhang Y, Fan Y, Song C, et al. Multi-agent rendezvous control based on event-triggered mechanism. In: *32nd Youth academic annual conference of chinese association of automation (YAC)*, Hefei, China, 19–21 May 2017, pp. 780–784. IEEE.
9. Zhao XW, Guan ZH, Li J, et al. Flocking of multi-agent nonholonomic systems with unknown leader dynamics and relative measurements. *Int J Robust Nonlin* 2017; 27(17): 3685–3702.
10. Luo X, Li X, Li S, et al. Flocking for multi-agent systems with optimally rigid topology based on information weighted Kalman consensus filter. *Int J Control, Autom Syst* 2017; 15(1): 138–148.
11. Zhang HT, Liu B, Cheng Z, et al. Model predictive flocking control of the cucker-smale multi-agent model with input constraints. *IEEE Trans Circuit Syst* 2016; 63(8): 1265–1275.

12. Zhou L and Li S. Distributed model predictive control for multi-agent flocking via neighbor screening optimization. *Int J Robust Nonlin* 2017; 27(9): 1690–1705.
13. Zhu L and Xiang Z. Aggregation analysis for competitive multiagent systems with saddle points via switching strategies. *IEEE Trans Neur Net Lear Syst* 2017; 29(99): 1–13..
14. Soysal O and Sahin E. Probabilistic aggregation strategies in swarm robotic systems. In: *Proceedings of IEEE swarm intelligence symposium, SIS 2005*, Pasadena, CA, USA, 8–10 June 2005, pp. 325–332. IEEE.
15. Garnier S, Jost C, Jeanson R, et al. Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. *Adv Artifi Life* 2005; 95: 169–178.
16. Guzel MS and Kayakökü H. A collective behaviour framework for multi-agent systems. In: Zhang D and Wei B (eds), *Mechatronics and robotics engineering for advanced and intelligent manufacturing*. Berlin: Springer, 2017, pp. 61–71.
17. Dorigo M, Trianni V, Sahin E, et al. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots* 2004; 17(2/3): 223–245.
18. Camazine S. *Self-organization in biological systems*. Princeton: Princeton University Press, 2003.
19. Ferrante E, Turgut AE, Duéñez-Guzmán E, et al. Evolution of self-organized task specialization in robot swarms. *PLOS Computat Biol* 2015; 11(8): 1–21.
20. De Luca A, Oriolo G, and Vendittelli M. Control of wheeled mobile robots: an experimental overview. In: Nicosia S, Siciliano B, Bicchi A, et al. (eds), *Articulated and mobile robotics for services and technologies*. Berlin: Springer Berlin Heidelberg, 2001, pp. 181–226.
21. Aicardi M, Casalino G, Bicchi A, et al. Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robot Autom Mag* 1995; 2(1): 27–35.
22. Aranda-Bricaire E, Salgado-Jimenez T, and Velasco-Villa M. Control no lineal discontinuo de un robot movil. *Comput Sist* 2002; (E):42–49.
23. Sahin E, Girgin S, Bayindir L, et al. Swarm robotics. In: Blum C and Merkle D (eds), *Swarm Intelligence*. Berlin, Heidelberg: Springer, 2008, pp. 87–100.
24. Soysal O, Bahceci E, and Sahin E. Aggregation in swarm robotic systems: evolution and probabilistic control. *Turk J Electr Eng Co* 2007; 15(2): 199–225.
25. Gauci M, Chen J, Li W, et al. Self-organized aggregation without computation. *Int J Robot Res* 2014; 33(8): 1145–1161.
26. Arvin F, Turgut AE, Bazyari F, et al. Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. *Adapt Behav* 2014; 22(3): 189–206.
27. Hu D, Zhong M, Zhang X, et al. Self-organized aggregation based on cockroach behavior in swarm robotics. In: *Sixth international conference on intelligent human-machine systems and cybernetics*, Hangzhou, China, 26–27 August 2014, pp. 349–354.
28. Arvin F, Turgut AE, Bellotto N, et al. Comparison of different cue-based swarm aggregation strategies. In: Tan Y, Shi Y, and Coello C (eds), *Advances in swarm intelligence*. New York: Springer International Publishing, 2014, pp. 1–8.
29. Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans Automat Contr* 2006; 51(3): 401–420.
30. Reynolds CW. Flocks, herds and schools: a distributed behavioral model. In: *Proceedings of the 14th annual conference on computer graphics and interactive techniques - SIGGRAPH '87*, New York, NY, USA, 1987, pp. 25–34. New York, NY: ACM Press.
31. Oh KK, Park MC, and Ahn HS. A survey of multi-agent formation control. *Automatica* 2015; 53: 424–440.
32. Tolba S, Ammar R, and Rajasekaran S. Taking swarms to the field: constrained spiral flocking for underwater search. In: *IEEE symposium on computers and communication (ISCC)*, Messina, Italy, 27–30 June 2016, pp. 1177–1184. IEEE.
33. Tanner H, Jadbabaie A, and Pappas G. Stable flocking of mobile agents. I. Fixed topology. In: *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, Maui, HI, USA, 9–12 December 2003, Vol. 2, pp. 2010–2015. IEEE.
34. Tanner H, Jadbabaie A, and Pappas G. Stable flocking of mobile agents. II. Dynamic topology. In: *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, Maui, HI, USA, 9–12 December 2003, Vol. 2, pp. 2016–2021. IEEE.
35. Altafini C. Some properties of the general n-trailer. *Int J Control* 2001; 74(4): 409–424.
36. Zebrowski P, Litus Y, and Vaughan RT. Energy efficient robot rendezvous. In: *Fourth canadian conference on computer and robot vision (CRV '07)*, Montreal, Québec, Canada, 28–30 May 2007, pp. 139–148. IEEE.
37. Wei R, Beard R, and Atkins E. A survey of consensus problems in multi-agent coordination. In: *Proceedings of the american control conference*, Portland, OR, USA, 8–10 June 2005, pp. 1859–1864. IEEE.
38. Wei R, Beard R, and Atkins E. Information consensus in multivehicle cooperative control. *IEEE Contr Syst Mag* 2007; 27(2): 71–82.
39. Lin J, Morse AS, and Anderson BDO. The multi-agent rendezvous problem. Part 1: the synchronous case. *SIAM J Control Optim* 2007; 46(6): 2096–2119.
40. Lin J, Morse AS, and Anderson BDO. The multi-agent rendezvous problem. Part 2: the asynchronous case. *SIAM J Control Optim* 2007; 46(6): 2120–2147.