Circle Formation in Multi-robot Systems with Limited Visibility

Eman Hasan[1(✉)], Khaled Al-Wahedi[1], Belal Jumah[1], Diana W. Dawoud[2], and Jorge Dias[1,3(✉)]

[1] Khalifa University of Science, Technology and Research, Abu Dhabi, UAE
emfhasan@pi.ac.ae

[2] Institute for Communication Systems, University of Surrey, Surrey, UK

[3] Institute of Systems and Robotic, University of Coimbra, Coimbra, Portugal
jorge@deec.uc.pt

Abstract. Pattern Formation in multi-robot systems was proposed in the 1990's. Since then it has been extensively studied and applied in various ways. To date, the majority of the proposed algorithms that aimed to achieve geometric patterns in the literature have overlooked the visibility limitation in physical robots. In addition, a methodology to reach a complete coordinate agreement has not been adopted by many researchers as a prerequisite towards a successful formation. It should be stressed that such limitation and methodology have a strong effect on the desired pattern approach. In this paper, a decentralized approach for circle formation is highlighted. The main advantage of forming a circle is the flexibility to be generated with different initial distributions. Moreover, circle arrangement can be utilized as a preliminary sub-task for more complex activities in multi-robot systems. To handle the aforementioned realities, this approach is proposed under a realistic robot model – i.e. one that has a short visibility range and performs the task autonomously relying on the information picked by itself, or by the vicinity. In addition, robots do not initially have a pre-defined leader nor unique IDs. Simulation results have validated the robustness and flexibility of the proposed algorithm, where a circular pattern has been successfully constructed in a self-organized manner.

1 Introduction

Over the last two decades, Swarm Robotics (SR) has been extensively investigated within the field of artificial intelligence. This extensive investigation has been strongly motivated by the presumed ability of a group of simple robots to accomplish a given complex task by distributing the work among themselves; bringing upon us a multi–agent system that is robust against member failure, and that is easily scalable. The research in SR finds its inspiration in the behavior of animals and insects; such as termites [2], bees, schools of fish, flocks of birds [6,10,15,28,29]. A group of simple robots moving as a swarm can achieve complex tasks such as collective mapping and searching, as wellas functioning

A. Ollero et al. (eds.), *ROBOT 2017: Third Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 693, https://doi.org/10.1007/978-3-319-70833-1_27

as a moving sensor array [21]. The main challenge in deploying large numbers of simple robots to accomplish a specific task is to find a strong framework to coordinate among the robots, especially in a distributed and decentralized fashion [8,11,13,16,19,23,27]. A strong coordination between the team members is achieved by defining a set of simple rules and interactions on the individual level to accomplish the desired global behavior. The research in multi-robot coordination tackles problems such as pattern formation, pattern movement and pattern switching. However, an efficient solution for the aforementioned problems has a strong dependency on the level of agreement on common coordinates among the robots. [7]. In line with this, a few number of studies investigated a unified approach in which the coordinates agreement was the leading step towards pattern formation [11,13,19,23]. This approach is adopted in this work.

The initial step for the swarm in accomplishing a task is to rearrange themselves from their initial random configuration into a specific spatial arrangement (e.g. a line, a wedge, a circle), and this step is referred to as pattern formation [4]. Therefore, it is important to secure an appropriate practical model for formation when apply it on a real word SR applications [20]. Visibility[1] is one of the main challenges in pattern formation, and it is assumed to be either unlimited or limited. In the former, the robots have the capability of sensing the whole plane [1,7,9,11,13,14,16,19,22,23,25,26,30]; while in the latter, the ability of robots to sense the other robots is limited to be within a certain range [3,17,24,26,30].

Although the majority of the prior work in the literature falls under the assumption of unlimited visibility, and it succeeds in forming the desired geometric pattern, that assumption is unrealistic due to sensor limitations. Furthermore, when the application requires the deployment of a large swarm of robots, the probability of robots blocking the line of sight of other robots increases. Capitalizing on that fact, the authors in [3,17,24,30] tackled the pattern formation problem in the form of gathering or convergence tasks under limited visibility. To this end, the only work that adopted a practical model of limited visibility and succeeded in forming a circular pattern was discussed in [26]. However, constant wireless communication is required among all robots, and an agreement on one common coordinate system is stated in advance.

To the authors' best knowledge, a solution, that adopts agreement on global coordinates system under a limited visibility assumption, to generate a circular pattern without requiring direct communication among all robots has not yet been achieved. Therefore, the main contribution of this paper is to propose a feasible algorithm for pattern formation while adopting a practical robot model. This practical model comprises the use of limited range sensors for measuring distance, and limited range transceivers for communication. Moreover, we do not require a pre-existing common coordinate system, nor do we require the robots to have pre-assigned leader nor external mark/ID. The only constraint that we have here is the requirement that the initial placement of robots in the environment guarantees that at least one robot has all other robots lying within

[1] Visibility refers to the way a robot observes the surrounding universe using its sensors. Robot's sensors take snapshots of the positions of all other robots within its visibility range with respect to its local coordinate system.

its visibility range. This assumption is necessary to ensure that all the agents in the team will eventually reach to a global coordinates agreement. Otherwise, the pattern formation problem for the proposed model will be unsolvable [13,26].

In the following section, the system model and main assumptions about robots capabilities are presented. Subsection 2.1 discusses the coordinate agreement algorithm for multi-robot system. A circle formation algorithm is described in Subsect. 2.2. Analysis and results are highlighted in Sect. 3. Conclusions are drawn in Sect. 4.

2 System Model

In this paper we consider a system, illustrated in Fig. 2, of n autonomous, anonymous, homogeneous, asynchronous and semi-obliviousness[2] robots, $r_1, r_2, \cdots, r_n$. The team members have an agreement on the unit distance and the orientation of positive x-axis (handedness/chirality), and are placed initially at random positions $p_1, p_2, \cdots, p_n$ as shown in Fig. 2(a). As already mentioned, the initial placement of the robots has only one requirement (constraint) and that is it must guarantee that at least one robot is able to see all other robots (consequently, it is also seen by all others). This is an essential requirement as we will see later. Each robot has its own local coordinate system, and it considers itself at the origin of that coordinate system. Furthermore, robot r_i using its local coordinates(denoted as L_i) can observe the robot r_j's position p_j if robot r_j is within robot r_i's visibility range as determined by its distance sensors. That is termed by $(L_i[x_j], L_i[y_j])$, where x_j and y_j are the x and y coordinates of p_j with reference to L_i. Moreover, r_i is assumed to be capable of detecting the positions of other robots in the system as well as broadcasting and receiving messages to/from the other robots that lie within its visibility range. In the following, the proposed coordinate agreement approach will be discussed in details in Subsect. 2.1. In addition, a circle geometric pattern will be demonstrated as an example of the proposed formation algorithm in Subsect. 2.2.

2.1 Coordinates Agreement Algorithm

To have a unified common coordinate system for a team of n robots, the stages shown in Fig. 1 have to be executed by each robot successively. Those stages are explained below:

Stage 1: Recognizing Team Members

In order to identify the number of robots in the team, each robot r_i has to go through the following steps:

[2] For non-oblivious robots used in [25], the authors assume that each robot is endowed with unbounded memory to store past information and execute a non–oblivious algorithm. However, since the robots in this work rely only on the most recent past information, it is required that each robot is equipped with a a limited amount of memory. Hence, the semi-oblivious nature of the algorithm [14].

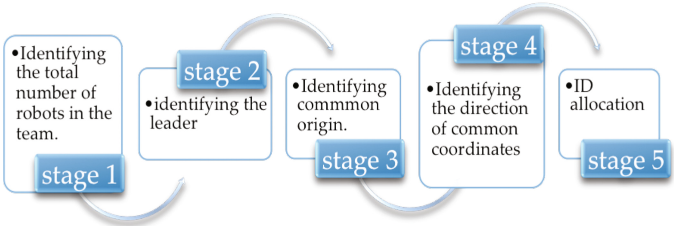


Fig. 1. Coordinate agreement stages

Step 1: r_i broadcasts and stores a "Hello message" that indicates the total number of detectable robots within its visibility range as can be seen by Fig. 2b.

Step 2: r_i receives a "Hello message" from robot r_j and process it according to the following: if the number indicated in the received "Hello message" is higher than the number detected by r_i, then r_i will replace the previously stored number with the newly received number, otherwise the message is discarded.

Step 3: r_i will remain in this broadcasting/listening mode for a predefined period of time[3]. At the end of this period, all robots will have identified the total number of robots in the group. Remember, we are requiring that at least one robot is able to see all other robots.

Stage 2 & 3: Electing a Team Leader & Common Origin

This work uses the leader-follower approach for robot formation, and thus the robots are required to identify a leader now. An intuitive approach is to select the robot with the most knowledge (i.e. the robot that sees all the other robots). That robot knows itself, so it can declare itself to the other robots with some specific flag (light, color, etc.). The other robots in their local coordinates mark the location of the leader r_l as p_l. The mechanism of leader election is illustrated in 2c. Moreover, in every robot configuration $C_i = L_i[p_k] \mid 1 \leq k \leq n$, p_l will be declared as the position of the common origin, $(0, 0)$, in the global coordinate system.

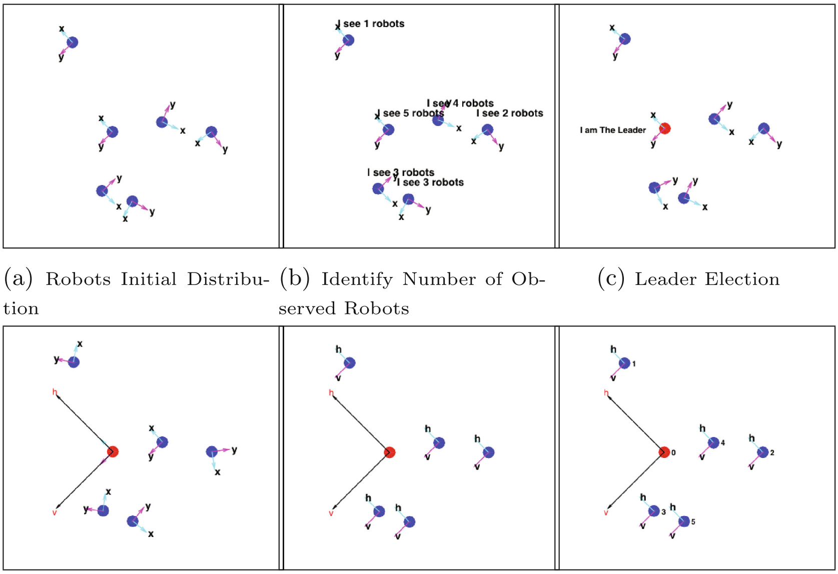
Stage 4: Direction of Common Coordinates

The common coordinates direction will be identified as indicated in the following steps:

Step 1: The local coordinates of the leader $(\overrightarrow{x_l}, \overrightarrow{y_l})$, which are characterized by the direction and orientation of the two axes, will be adopted as a common coordinates by the rest of the team members.

Step 2: Until this moment, the robots are not aware with the direction of the leader's coordinates. As a resolution, the leader will move one step forward in its positive $\overrightarrow{x}$-axis, then backward to its original position p_l (mimicking bees in a bee dance) [16]. This movement entices the other members attention to identify

[3] This period is determined so as to allow all robots to broadcast their "Hello message". A suitable negotiation mechanism can be easily devised to determine the order in which all robots broadcast.



(a) Robots Initial Distribution (b) Identify Number of Observed Robots (c) Leader Election

(d) Identify the system Global coordinates (e) Robots Coordintas Rotation (f) ID Allocation

Fig. 2. Coordinate agreement and ID allocation

the movement's direction as the horizontal axis of the common coordinate frame and term it as ($\overrightarrow{h}$-axis).

Step 3: The direction of the vertical axis ($\overrightarrow{v}$-axis) is determined by rotating the horizontal $\overrightarrow{h}$-axis 90 degrees counterclockwise. Common coordinates directions are outlined in Fig. 2d.

Step 4: Each robot r_i is required to align its local coordinates system to match the global coordinates system as in Fig. 2e. This alignment can be easily accomplished using a simple rotation transformation.

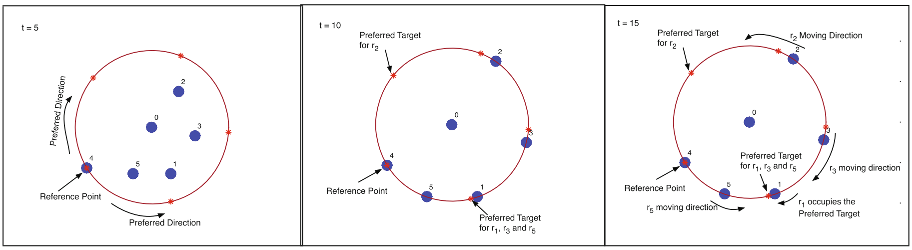
Stage 5: ID's Allocation

In the ID allocation mechanism, the leader assigns himself "0" ID. It then broadcasts $(n-1)$ messages one after the other to the remaining $(n-1)$) robots. Each of these messages contains a location in the global frame followed by a number (ID). Each robot receives the message, and then compares its location with the location in the message. If they match, it assigns itself the ID contained in that message; if not, it discards the message. Figure 2f illustrates the ID allocation mechanism. By the end of stage 5, the robots now share one common coordinates system, have elected a leader and have their unique IDs. Therfore, they are now ready to form geometric pattern as specified in [7,12,19,25].

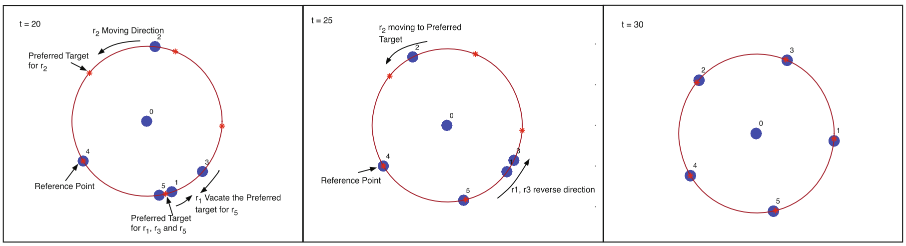
2.2 Circle Formation Algorithm

Circle formation is defined as the coordination between a group of robots to get and maintain a circle shape [4]. One of the major advantages of forming a circle is the flexibility to be generated with different initial distributions of robots and group sizes. Circle shape has different applications, such as autonomous surveillance, encircles the enemy's targets, protects moving convey and rescue missions [5]. In this paper the goal is to create the desired circular shape as independently as possible (i.e. with the minimum amount of interactions among team members). To avoid collisions among robots, the proposed approach to the circle generation task will be divided into two main stages: first, to reach the circle circumference; and second, to distribute the robots uniformly along the circle's circumference. Both stages are discussed in details in the following subsections.

Reaching the Circle's Circumference. In the proposed circle algorithm, the leader will be the center of the circle and will stay stationary. It is the leader's responsibility to inform other members with the desired pattern, which is a circle shape in our scenario. Moreover, the leader would broadcast the coordinates of the farthest robot with respect to its position. In the case, where there is more than one robot that has the same farthest distance from the leader, the leader will choose the robot with the lower ID as a reference point, $p_{ref} = (h_{ref}, v_{ref})$. This



(a) Select Reference point (b) Determine Preferred Target Point (c) Competing on the Preferred Target



(d) Reverse Direction (e) Competing on the unclaimed Target Points (f) Complete Circle Formation

Fig. 3. Circle formation as time passes

point will be later used to determine the radius (R) of the circle, by estimating the distance between p_l and p_{ref}. Note that this desired circle contains all the robots inside it or on the circumference.

Figure 3a illustrates the chosen reference point. With this limited amount of (i.e. p_{ref}), the remaining robots are able to determine the radius R of the circle, and also determine their target locations along the circumference. Here, we define the target locations as a set of points dividing the circle's circumference into equal parts. The number of target points depends solely on the number of robots in the team. Based on the computed R, r_i enters an iterative loop to check the distance with r_l while moving radially outward in small steps until it reaches the circle circumference. Note that our original constraint that the leader maintains a line of sight with all other robots comes in handy here, because each robot can move radially outward without worrying about colliding with other robots (see Fig. 3a). This stage will be complete when all robots are on the circle's circumference. At which point, the leader broadcasts a message that all robots have reached the circumference as shown in Fig. 3b. We then move to the second stage.

Distribute Uniformly Along Circle's Circumference. The goal of this stage is for the robots to uniformly distribute themselves along the circle circumference

Step 1: Determine robot's preferred direction

As a first step, each robot determines in which half of the circle it is located, with respect to the line passing through p_{ref} and the center of the circle p_l. Based on that, the robot's preferred direction is defined as the direction going from reference point to the nearest target points as shown in Fig. 3(a).

Step 2: Determine the preferred target point

All robots in each half of the circle are aiming for the same target point, which is the nearest target point to the reference point in that half.That implies that more than one robot will get the same point as its preferred target point. Figure 3b shows the preferred target point for the robots in the system.

Step 3: Move to the intended target point:

Now all robots will start moving asynchronously towards the preferred target point as illustrated in Fig. 3c. The robot that will first reach the preferred target will occupy it. Later, it either keeps it or leaves it according to the strategy explained in the following subsection.

Competing on Target Points Strategy. Since robots are required to distribute themselves uniformly along the circle circumference, a smart moving strategy has been devised to guide robots to their target points. This subsection will shed the light on the competing strategy, which represents the core of this work. Beside identifying the preferred target point, each robot requires avoiding collision with other robots in the team. Therefore, the robot r_i needs to find a powerful technique to reach its target point. The following simple rules are

Algorithm 1. Circle Pattern Generation

Input: Each robot has ID_i, the leader global position (h_l, v_l). Also, the reference point coordinates (h_{ref}, v_{ref}) and the required pattern shape is broadcast-ed by the leader

Output: Circle Pattern Formation

1 Compute R

Find,

$\Delta_x = x_{(refpoint)} - x_{(leader)}$.

$\Delta_y = y_{(refpoin)} - y_{(leader)}$.

$R = \sqrt{\Delta_x^2 + \Delta_y^2}$

2 Move to Circle Circumference,

Check,

2.1. if distance $(r_l, r_i) = R$. then, do not move.
else

2.2. if distance $(r_l, r_i) < R$. Move radially outward for a small step

2.3. Repeat *Steps 2.1 to 2.2*, till $|distance(r_l, r_i)| = R$

3 Compute Target Points.

$\theta = \frac{2*pi}{n-1}$

$D = [1\ 0\ 0] \cdot [\frac{\Delta_x}{R}\ \frac{\Delta_y}{R}\ 0]$

$C = [1\ 0\ 0] \times [\frac{\Delta_x}{R}\ \frac{\Delta_y}{R}\ 0]$

$\phi = \arctan(\frac{C}{D})$

for *j = 1: n-1;*

$Targetpointsh_i = [h_l + R * cos(\phi + \theta * j)]$

$Targetpointsv_i = [v_l + R * sin(\phi + \theta * j)]$

$j = j + 1;$

end

f_i = (Target points h_i,Target points v_i)

4 Move to Target Point Using Competing on Target Points Strategy

suggested as a way for indirect communication between any two robots in the team:

1. Rule 1: If the robots r_i, r_j move in the same direction and opposite to preferred direction, the first robot that occupies the target point has the priority to stay there.
2. Rule 2: If robots r_i, r_j move according to the preferred direction, the robot which occupies the target point first should leave the target point to the second robot and start looking for the nearest unclaimed target point.
3. Rule 3: If robots r_i, r_j move opposite to each other and aiming for the same target point, the robot which moves in the preferred direction has the priority to get the intended target point. The other robot must inverse its direction looking for the next unclaimed target point.

Figure 4 illustrates the possible states for r_i, r_j and the rules explained above.

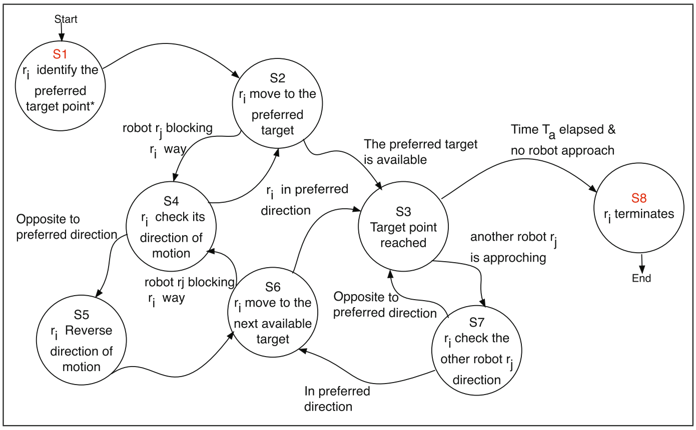


Fig. 4. Finite state machine

When all robots in the team apply the rules in the finite state diagram, the robots will be able to accomplish the task of circle generation in a self-organized manner using the limited information provided earlier by the leader. The complete circle generation algorithm is explained in details in (1).

3 Analysis and Results

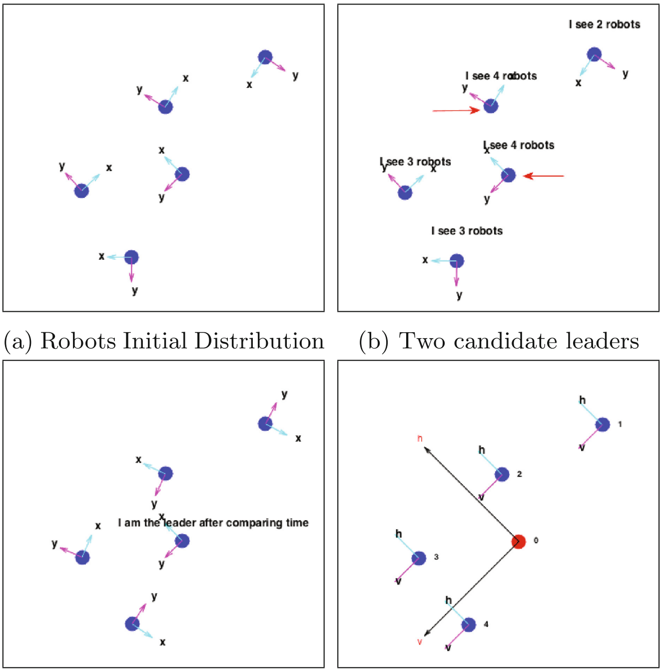
3.1 Multiple Leader and Robot Loss

In some situations, which depend on the robots' initial distribution, two or more robots could be candidates to play the leader role. All robots need to walk through the same mechanism discussed in (Sect. 2.1) to elect only one leader. However, the first robot that identifies the largest number of detected robots using the broadcasted message will be assigned as the leader for the system. Multiple leader selection mechanism is illustrated in Fig. 5.

Furthermore, if one of the members fails to reach the circle circumference due to a physical damage, the leader will acknowledge the team members with the number of remaining robots. Accordingly, the team members will start again in executing the above algorithms.

3.2 Factors Affect the Algorithm Performance

Our approach utilizes many agents which use the local information to produce a circular shape in a decentralized manner. The main question that arises is whether these local actions always generate the desired shape form its initial states. Moreover, what are the factors that affect the system convergence. By leveraging the results form Matlab simulations, it was proven that the circle algorithm converges to the desired circle for different initial distributions and



(a) Robots Initial Distribution (b) Two candidate leaders

(c) Leader Selection (d) ID Allocation

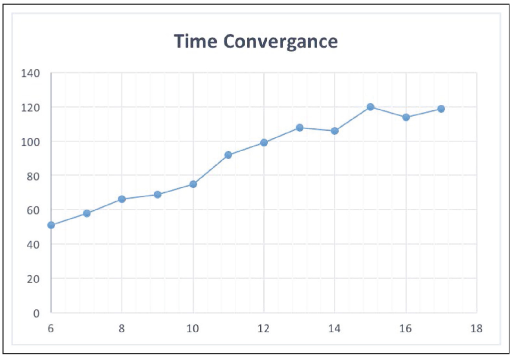
Fig. 5. Multiple leader selection mechanism

various number of agents in the team. Two factors are addressed to measure the time convergence (the time required to achieve the required task): (1) Robot's visibility range and (2) The number of robots in the team. The following results are derived from vast number of simulations: Results in Fig. 6a show how the time required to achieve the circle shape increases when the robot's visibility range increases for the same team size. This implies that increasing the agent's distance sensors ranges would allow the robots to scatter and navigate in a larger space and hence require the robots to incur more time in aggregating the required information to complete the task. However, increasing the robot's range would provide a better coverage for the surrounding universe [20].

Observations in Fig. 6b reveal that the convergence rate scales with the number of robots in the team. While the visibility ranges for all agents are equal, the time required to generate the circle increases when the number of robots in the team increased. Moreover, Robors' initial states play an important role in time convergence. To illustrate, it occur that a team of 13 robots spread out initially in the environment in a way demands more steps, which incur more time, to complete the circle in comparing with a team of 14 robots as shown in Fig. 6b.

Visibility Range	10	15	20	25
Time Convergence	60.4	71.4	77	82.3

(a) Robots' Visibility Range



(b) Number of Robots

Fig. 6. Factors affect time convergence

3.3 Metric Analysis

The proposed algorithms (coordinate agreement and circle formation) were simulated in an ideal environment (assuming no noise). The simulation is carried out using Matlab. Figures 2 and 3 are both examples of simulation results for a team of 6 robots. The proposed algorithms worked successfully for different initial distributions of robots, and can be scaled to include different group sizes. Furthermore, it should be highlighted that the circle's radius would vary according to the reference point's location with respect to the leader regardless of the number of robots in the team.

In this paper, we tried to propose a new solution for an existing problem that has been discussed by many researchers. Therefore, an analytical comparison between this work and other well-known studies in the literature [9,19,23,25,27] will be used to to evaluate the performance of the proposed algorithm. This comparison is carried out based on the robot's model, assumptions and the designed circle algorithm.

Lee and Chong work in [19] depends strongly on system's visibility to generate the formation pattern and it assumes that the robots are able to detect all other robots' positions in the field (unlimited visibility), which makes their system impractical for hardware implementation. In their work, to generate a pattern, the robots in [19] are required to have symmetrical distribution around the x-axis of the common coordinates based on their IDs, which makes their system non-flexible for physical implementation. Whereas, this work do not rely on robots IDs or distribution to generate the circle pattern, which will make the proposed algorithm more practical.

Karthikeyan in [27], adopted the model in [25]. However, they assumed that the robot can sense the position of other robots within a certain visibility distance, which makes the model more feasible. Moreover, the robots are equipped with transmitter/receiver to broadcast/receive the members' positions instead of sensors used for direct observation. On the other hand, the robots in this work are assumed to use both equipments to detect other robots' positions and broadcast/receive messages as well. The robots in [27] have unbounded memory (non-oblivious) and they are assumed to share the same coordinates system in prior. Interference between broadcasted/received messages could affect the system performance and stability.Whereas, the proposed algorithm in this work tried to develop a set of rules that enable the robots to accomplish the circle formation in self organized manner.

However, several issues, such as optimizing the number of steps to generate the desired pattern, improving the system efficiency by reducing the delay that could happen while broadcasing/receiving messages and generating more geometric patterns, remain open and will be addressed in future works.

3.4 Decentralized Strategy and Centralized Planner

It is considered by some works that working in multi-agents systems should be either using a decentralized strategy, in which all agents repeatedly communicate with each other and with the environment to achieve the given task, or there should be a central eye in the system that aggregate all the required information and distribute them among the team's members. Nevertheless, centralized strategy has many limitations [18]. The approach adopted in this work tried to find a trade-off between these two approaches. Our system uses the leader-follower approach, which considers the leader as a median agent (more than leader). Despite the fact the leader presence is vital to maintain a more robust system, its roles are very limited. Furthermore, keeping bonds between leader and team members could help in keeping the generated pattern as a rigid structure. In our approach, using the leader to broadcast/ receive information from all other team's member would reduce communication cost in comparison with the case of all agents would broadcast/receiveinformation with and/or around each other [31]. Therefore, a decentralized strategy is used, in which the agents in the team require to collect information from the surrounding environment and control its state based on that.This adaptation helps later to solve effectively all scenarios using a distributed approach. In addition to that, a centralized planner is necessary to oversee and monitor all decentralized tasks.

4 Conclusions

A study on simple robots that are equipped with simple sensors and tools to detect the nearby robots has been presented. A unified algorithm has been demonstrated for coordinate agreement among a team of mobile robots, enabling them to consent to a common coordinate system. The robots can execute the proposed algorithm under practical conditions such as: Non existence of a pre-assigned leader or IDs. Only a few researchers have included the coordinate agreement as an essential step toward pattern formation. Others have assumed that the coordinate agreement has already been reached, and focused on patterns generation. Another larger group of researchers devised their pattern formation algorithms based on the unrealistic assumption that the robots have unlimited visibility. In this work a practical model is presented where the limited visibility assumption has been adopted to generate a specific geometric pattern. In addition, an elegant algorithm that enables a team of robots to work cooperatively and generate a circle pattern in a decentralized manner was developed. Furthermore, the algorithm used in this work utilizes a set of simple rules and behaviors, which permit each robot to accomplish its task independently without relying on any direct negotiation with the other team members. Simulation results proved the feasibility and scalability of the algorithm for a large number of robots.

References

1. Aguilera, M.K., Querzoni, L., Shapiro, M. (eds.): Principles of Distributed Systems. LNCS, vol. 8878. Springer International Publishing, Cham (2014)
2. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., Beardsley, P.: Multi-robot system for artistic pattern formation. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4512–4517 (2011)
3. Ando, H., Suzuki, I., Yamashita, M.: Formation and agreement problems for synchronous mobile robots with limited visibility. In: Proceedings of 10th International Symposium on Intelligent Control, pp. 453–460, August 1995
4. Bahceci, E., Soysal, O., Sahin, E.: A review: pattern formation and adaptation in multi-robot systems. Technical report CMU-RI-TR-03-43, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2003)
5. Barca, J.C., Sekercioglu, A.: Generating formations with a template based multi-robot system. In: Australasian Conference on Robotics and Automation, Melbourne, pp. 1–9 (2011)
6. Barca, J.C., Sekercioglu, Y.A.: Swarm robotics reviewed. Robotica **31**(03), 345–359 (2012)
7. Chatzigiannakis, I., Markou, M., Nikoletseas, S.: Distributed circle formation for anonymous oblivious robots. Exp. Eff. Algorithms **33116**, 1–16 (2004)
8. Chaudhuri, S.G., Ghike, S., Jain, S., Mukhopadhyaya, K.: Pattern formation for asynchronous robots without agreement in chirality. CoRR abs/1403.2625 (2014)
9. Défago, X., Souissi, S.: Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. Theor. Comput. Sci. **396**(1–3), 97–112 (2008)
10. Edelen, M.R.: Swarm intelligence and stigmergy: robotic implementation of foraging behavior. Master's thesis, University of Maryland (2003)
11. Flocchini, P., Prencipe, G., Santori, N., Widmayer, P.: Hard task for weak robot: the role of common knowledge in pattren formation autonomous mobile robots. In: 10th International Symposium, ISAAC 1999, Chennai, India, vol. 1741, pp. 93–102 (1999)
12. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Morgan & Claypool, San Rafael (2012)

13. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Pattern formation by autonomous robots without chirality. In: Proceedings of the 8th International Colloquium on Structural Information and Communication Complexity, SIROCCO, vol. 8, pp. 147–162 (2001)
14. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. Theor. Comput. Sci. **407**(1–3), 412–447 (2008)
15. Gazi, V.: Swarm aggregations using artificial potentials and sliding-mode control. IEEE Trans. Robot. **21**(6), 1208–1214 (2005)
16. Gordon, N., Wagner, I., Bruckstein, A.: Discrete bee dance algorithm for pattern formation on a grid. In: IEEE/WIC International Conference on Intelligent Agent Technology, IAT 2003, pp. 545–549 (2003)
17. Hu, J., Feng, G.: Distributed tracking control of leader-follower multi-agent systems under noisy measurement. Automatica **46**(8), 1382–1387 (2010)
18. Khaldi, B., Cherif, F.: An overview of swarm robotics: swarm intelligence applied to multi-robotics. Int. J. Comput. Appl. **126**(2), 31–37 (2015)
19. Lee, G., Chong, N.Y.: Decentralized formation control for a team of anonymous mobile robots. In: Proceedings of 6th Asian Control Conference, pp. 971–976 (2006)
20. Liekna, A., Grundspenkins, J.: Towards practical application of swarm robotics: overview of swarm tasks. In: Proceedings of the 13th International Conference on Engineering for Rural Development, pp. 271–277 (2014)
21. Navarro, I., Matía, F.: A survey of collective movement of mobile robots. Int. J. Adv. Robot. Syst. **10**, 73 (2013)
22. Prencipe, G., Santoro, N.: 4th IFIP International Conference on Theoretical Computer Science- TCS 2006. nt. Federation for Inform. Process. Springer, US, Boston, MA (2006)
23. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. J. Robot. Syst. **13**(3), 127–139 (1996)
24. Suzuki, I., Yamashita, M.: A theory of distributed anonymous mobile robots formation and agreement problems. Technical report ADA303910, Wisconsin Univ-MilwaukeeI Dept. of Elect.l Eng. and Compu. Sci. (1994)
25. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: formation of geometric patterns. SIAM J. Comput. **28**(4), 1347–1363 (1999)
26. Swaminathan, K.: Self-organized formation of geometric patterns in multi-robot swarms using wireless communication. Master's thesis, University of Cincinnati (2005)
27. Swaminathan, K., Minai, A.: A general approach to swarm coordination using circle formation, pp. 65–84. Springer, Berlin, Heidelberg (2006)
28. Tan, Y., Shi, Y., Ji, Z. (eds.) Advances in Swarm Intelligence. LNCS, vol. 7331. Springer, Berlin, Heidelberg (2012)
29. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science **343**(6172), 754–8 (2014)
30. Yamauchi, Y.: A survey on pattern formation of autonomous mobile robots: asynchrony, obliviousness and visibility. J. Phys. Conf. Ser. **473**(1), 012,016 (2013)
31. Yu, C.H., Nagpal, R.: Biologically-inspired control for multi-agent self-adaptive tasks. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI) (2010)