Simulation of Pattern Formation of Swarm withMinimum Shape Parameters

Md. Tahmeed Abdullah*, Md. Jubair Ahmed Sourov†, Sejuti Rahman‡ and Sujan Sarker§
Department of Robotics and Mechatronics Engineering
University of Dhaka, Dhaka-1000, Bangladesh
*a.tahmeed@yahoo.com, †sourovejubair@gmail.com, ‡sejuti@gmail.com, §sujan@du.ac.bd

***Abstract*—The topic of initializing and controlling the formation of robots in a multi-agent system has become a core swarm robotics research problem as formation control is very useful for cooperative tasks. In this paper we propose a hybrid method for formation control of non-holonomic mobile robots fusing controller behaviours with virtual structure. The proposed method is ideal for applications like search and rescue operations. The method is tested and evaluated in some simulation experiments and it proved to perform better for random shapes as well as polygonal shapes and convex hulls than behaviour based approaches and it was observed that converging time is independent of the shape complexity. The method also provides an easy method to design new pattern shapes. The results show that the proposed method improves in accuracy, formation controlling mechanism, applicability and variation of shapes.**

***Index Terms*—swarm, formation, centralized, hybrid, behaviour based, virtual structure**

I. Introduction

In recent years, *swarm robotics* has emerged as the application of swarm intelligence into multi-robot systems, which focuses on the physical embodiment and realistic interactions into robot to robot and robot to the environment. Inspired by the compelling collective behavior of social animals, swarm robotics systems has the following advantages [1] : *Robustness*, the system is able to operate in spite of disturbances from environment or failure of individuals; *Parallel*, each agent of the system simultaneously carry out the task; *Scalable*, as the agents only interact locally, any individual can join or quit the system without affecting the whole system performance.

In swarm robotics applications like search and rescue, surveying, exploration, the initial formation of robots is essential. The initial formation of robots formed from scattered positions is done by aggregation of robots or rendezvous of robots at some points in space followed by forming the desired shape.

The formation control is concerned with the coordination of a group of robots in maintaining a particular position while moving through an environment [2]. This problem has drawn the attention of many researchers recently and various schemes or approaches to solving this problem have been proposed. According to the survey in [3], all the existing methods can be classified into the following three approaches-

- *Behaviour-based approach*: The collective swarm behavior is achieved by a mathematical weighted function of multiple simple behaviors such as avoid robots, avoid obstacles, follow walls, goal-seeking, etc [4]. This method is decentralized and can be implemented with less communication.
- *Leader-follower approach*: A leader robot is selected by the operator or by the swarm itself by means of sophisticated algorithms and the chosen leader robot maintains a reference position or follows a trajectory set dynamically or by an operator. The followers track a relative distance and orientation with the designated leader robot.
- *Virtual structure strategy*: Virtual structure method was first introduced in 1997 [5]. The entire formation of robots is considered as a structure and the robots are perceived as joints of that moving structure.

In terms of control architecture, swarm robotics systems are either centralized or decentralized. In decentralized swarm robotics systems, the robots interact locally and eventually take and execute decisions autonomously. Decentralized architectures promise scalable systems. But according to [6], for a limited number of robots, centralized swarms outperform decentralized systems in any algorithm. In [7], a scalable decentralized pattern forming algorithm is proposed, where the robots use bearing information about the shape to generate that shape.

Numerous self-organization, pattern formation, and formation control mechanisms are inspired by mathematical modeling of physical phenomenons like spring-damper system [8] with PD or PID controllers [9] where two robots in the system are connected by a virtual spring-damper mechanism or by smooth hydrodynamic behavior [10]. Constants like spring stiffness, damping coefficient, and angular velocity constant of any two robots need to be hand-tailored and requires fine-tuning of parameters to build a steady system.

The proposed simulated work in this paper is designed for a heterogeneous swarm of robots where there are layers of command hierarchies. An operator or central command station communicates directly with an aerial robot (e.g. a quad-copter) which monitors, receives feedback, and commands a group of mobile robots to generate patterns. Hence, the system is robust as the mobile robots are not fully dependant on the central control system or the operator. If the monitoring aerial robot fails or gets damaged, another aerial robot can take its charge and take care of the situation. If the central control system fails to operate, the aerial robot keeps doing its task until the operator comes back online.

The proposed work also achieved to form the accurate shape by a swarm of simulated robots by a minimum number of shape-specific parameters. The least number of parameters required to fully control the characteristics of a shape is taken into account.

Among all the drive systems of mobile robots, the differential drive system is the most popular and common as it is easier to model in real-life scenarios. Our simulation is generalized for all non-holonomic mobile robots including differential drive robots.

The contributions of our paper are-

- Design of a formation control strategy that requires minimum shape-specific parameters which makes it easier to implement.
- Development of a new and easy strategy to develop simple patterns by developing simple equations or through images.
- Development of an algorithm for target point assignment to robots for which the converging time to the desired shape is independent of the shape complexity.

II. Methods

This section discusses the methods we followed to simulate a swarm of robots to create patterns. First, the simulation software and setup are discussed. Then the design of robot controllers using the simulation setup is discussed. Finally, some evaluation methods are discussed to determine the performance of the swarm of robots based on our proposed algorithm.

A. Simulation Environment

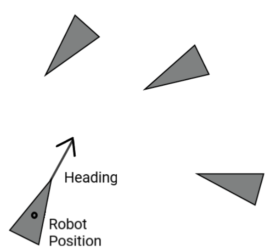


Fig. 1: Triangular boids represent the autonomous robots in our simulation where the vertex is the heading of the robot and the centroid of the triangle denotes the position of the robot with respect to some coordinate.

The evaluation of the proposed method was done by some experimentation which was performed on a graphical programming software called 'Processing' [11] using Java programming language. A simulation environment is designed by the authors which are available in Github[1]. The robots are designed inspired by Reynold's flocking and steering simulation of 'Boids' [12]. These boids are triangle-shaped vehicles which are placed in a 2D canvas.

[1]Github Link: https://github.com/atahmeed/swarm_formation_simulation

In this simulation, the state of a boid is defined by 4 parameters- position in 2-dimensional space, heading angle, velocity and acceleration. Each boid is assumed to have unit mass. For a given non-zero acceleration $\vec{A}$, velocity $\vec{V}$ and $\vec{X}$ for the position of a boid, in simulation is updated at each time step t due to application of force $\vec{F}_{steer}$, by the equations given in Eq.1-4. Note that, each time step in the simulation is measured by the time to complete each iteration in the simulation loop.

For a given maximum speed $maxV$,

$$\vec{A} = \vec{A} + \vec{F}_{steer}\Delta t \tag{1}$$

$$\vec{V} = \vec{V} + \vec{A}\Delta t \tag{2}$$

$$\vec{V} = min(\vec{V}, maxV) \tag{3}$$

$$\vec{X} = \vec{X} + \vec{V}\Delta t \tag{4}$$

The force on the robots is set by the behavior or the environment. For goal-seeking behavior, a steering force is applied to set the heading of the robot to the direction of the goal and for collision-avoidance behavior, the force is applied to the robot to set heading away from the colliding agent.

B. Robot Controller

The robot controller is designed based on the combination of some control behaviors. In fact, the output velocity of the robot controller is the weighted sum of the output velocity of each behavior controller which is represented mathematically in Eq.5.

$$\vec{V}_{Desired} = \sum_i W_i \vec{V}_i \tag{5}$$

The W_is are the design parameters for each behavior which generally range from 0 to 1 and $\vec{V}_i$s are the output of each behavior.

In this paper, only two simple behaviors configure the dynamics of robot motion and interaction: "Goal-seeking" behavior and "Avoid-obstacles" behavior.

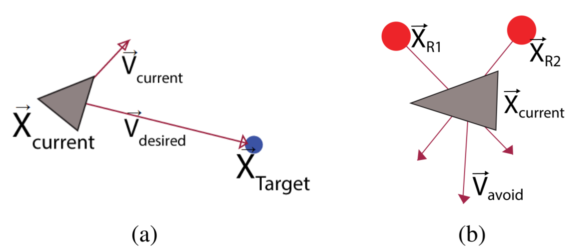


Fig. 2: (a) The blue circle denotes the target position and the goal seeking velocity of the robot is determined by the Goal-seeking behaviour. (b) The red circles denote the obstacles/robots and the avoiding velocity is determined by the Avoid obstacle behaviour.

1) Goal-seeking behaviours
The "Goal-seeking" behavior, as depicted in Figure 2a, makes a robot always reach a specific goal in the geometric space. To design a proportional controller for the goal-seeking behavior for a robot at position $\vec{X}_{current}$ with a target coordinate at $\vec{X}_{target}$, let us assume the current velocity and desired velocity of the robot for this behavior to be $\vec{V}_{current}$ and $\vec{V}_{goalseek}$ respectively. The steering force applied to the robot to achieve $\vec{V}_{goalseek}$ is determined by the equations in Eq. 6-8.

$$\Delta \vec{X} = \vec{X}_{target} - \vec{X}_{current} \tag{6}$$

$$\angle \vec{V}_{goalseek} = \angle \Delta \vec{X} \tag{7}$$

$$|\vec{V}_{goalseek}| = |\angle \Delta X| \times K_p \tag{8}$$

K_p is the proportional constant (value depends on controller designer) and the operators Δ, $\angle$, $|\;|$ denote *difference*, *angle* and *absolute value* respectively.

2) Avoid-obstacles behaviours
The robot moves 'away' from the objects (e.g. other robots, walls, obstacles etc.) it is trying to avoid. In other words, the robot moves in a direction that is opposite(180°) to the direction of the obstacles/robots to avoid a collision. For multiple objects/robots within the close vicinity of the robot, the robot goes in a direction which is the summation vector of all the vectors opposite from each robot/object in the vicinity as shown in Figure 2b.For all robots, $R_1, R_2, R_3, .., R_N$, the desired avoid obstacle controller output is determined by Eq. 9.

$$\vec{V}_{avoid} = \frac{1}{\Delta t} \sum_{i}^{N} \vec{X}_{current} - \vec{X}_{Ri} \tag{9}$$

Fusing these two behaviours using Eq. 5, we get the desired velocity and steering force $\vec{F}_{steer}$ (Eq. 10-11) which is fed into boid controllers and the boids act upon the steering force using the equations in Eq. 1-4.

$$\vec{V}_{desired} = W_1 \vec{V}_{goalseek} + W_2 \vec{V}_{avoid} \tag{10}$$

$$\vec{F}_{steer} = \frac{K}{\Delta t} \times (\vec{V}_{desired} - \vec{V}_{current}) \tag{11}$$

III. Simulation Experiments

For any simulation, we assume to have a set of robots $R_1, R_{,}.., R_N$ and a set of target points $T_1, T_2, .., T_M$, where N and M denote total number of robots and target points respectively. The robots can seek any target point and occupy that point following the Algorithm **??**.

Iterating through the robots, each robot is assigned with a target point. At first, the monitoring aerial robot checks whether a robot is assigned to a specific target point or not

Algorithm 1 Formation Control algorithm for predefined set of target points

while each Robot is not settled **do**
  for $R_i : i = 0 \rightarrow N$ **do**
    if R_i is not assigned **then**
      $T_i^{R_i}$ = closest target-point for R_i
      if $T_i^{R_i}$ is occupied **then**
        continue
      else
        assigne R_i to $T_i^{R_i}$
      end if
    else
      R_i seeks target point $T_i^{R_i}$
    end if
  end for
end while

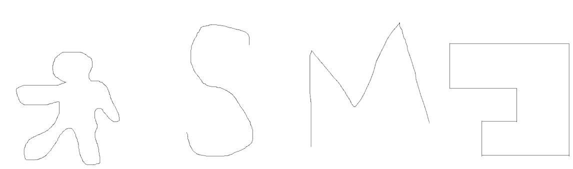
and the nearest unoccupied target point of an unassigned robot is assigned to it. The robot then seeks that target point.

How do we know if each robot is settled or not? We measure an error at each time step for each robot and determine an average error, E_{avg}. If the average error reaches a near zero value, the robots have settled down to their desired target points.

$$E_{avg} = \sum_{i}^{N} \frac{1}{N} \times (T_i^{R_i} - X_{R_i}) \tag{12}$$

The generation of target points is studied in two experiments in two different ways. In the first experiment the target points are generated from a 2D binary image whereas, in the second experiment, the target points are generated automatically depending on the number of robots and some defined shape-specific parameters.

Experiment 1: Formation of swarm pattern by generating target points from images For a given input of 2D image from the central system, the monitoring robot create as set of equally spaced target points from the binary image assigns each robot with a target point and the robots seek that point. Figures 3(a-d) show the input images to the system and we shall denote these shapes as (a) Human shape (b) S shape (c) M shape (d) Random shapes respectively.



a b c d

Fig. 3: Binary images fed into the swarm system for experiment 1. (a) Human shape (b) S shape (c) M shape (d) Random

Experiment 2: Formation of swarm pattern by automatically generating target points with minimum shape-specific parameters

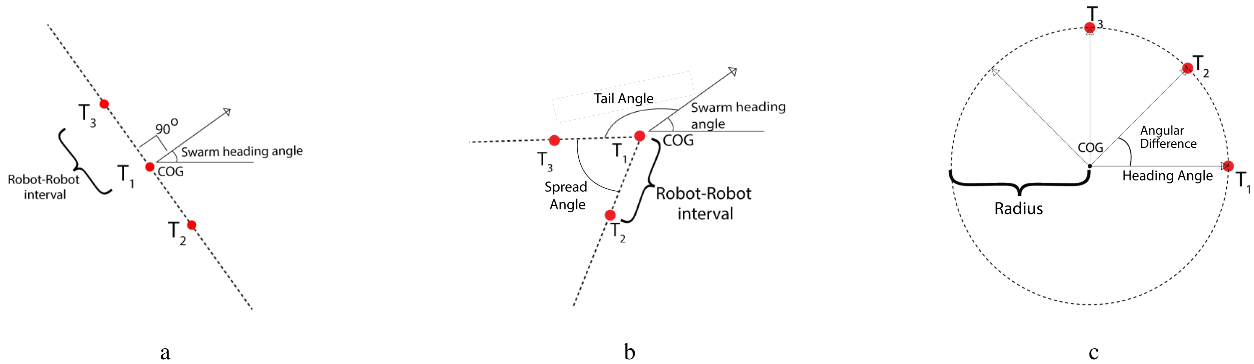


Fig. 4: Automatic generation of target points (a) on straight line given three shape parameters COG, heading angle and length of the line,(b) on arrowhead given four shape parameters COG, heading angle, spreading angle and length of the lines and (c) on the circumference of a circle given three shape parameters COG, heading angle and radius of the circle for experiment 2

Three types of shapes are designed and the corresponding swarm patterns are formed. The shapes are:

1) **Straight Line** To generate a straight line with a given number of robots, the system needs three information, the coordinate of the line's center of gravity(COG), the heading angle of the swarm and length of the line. Starting from the COG. The system generates target points at definite intervals at two sides of the COG perpendicular to the direction of swarm heading. Note that, interval length is determined by the length of the line segment and the number of robots in the swarm. The purpose of each parameter is provided in Table I.

TABLE I: Shape parameters

Shape	Parameter	Purpose/Function
Straight Line	COG	Locate middle point of the straight line
	Length	Determine the length of the straight line
	Heading Angle	Determine angular orientation of the straight line
Arrowhead	COG	Locate the tip of the arrowhead
	Spreading Angle	Determine the angle between the tails of the arrowhead
	Length	Determine the length of each tail
	Heading Angle	Determine Angular orientation of the arrowhead
Circle	COG	Locate middle center of circle
	Radius	Determine the radius of the circle
	Heading Angle	Determine the direction of the swarm

2) **Arrow head** This shape is a more generalized version of straight-line. Instead of placing target points making a 90°angle with the swarm heading angle, the value of this angle depends on a new parameter, 'Spread angle'. The spread angle is the angle between the two tailing line segments of the arrowhead. In Figure 4b the spread angle is used to determine the 'TailAngle', which is the angle between the heading and the line-segment.

3) **Circle** The target points in the circle are added at a definite angular difference from the swarm heading angle. Instead of the "spread angle", a new parameter "radius" is used to determine the radius of the circle to be formed by the robots. Therefore, the target points equal to the number of robots in the swarm are placed around the COG at a distance of "radius" and at an angular interval of "angle difference" which is calculated from the number of robots.

It is worth noting that the parameters for each shape described in Table I control all the characteristics of that shape fully. The control of each of these characteristics cannot be done with fewer parameters.

IV. Experimental Results

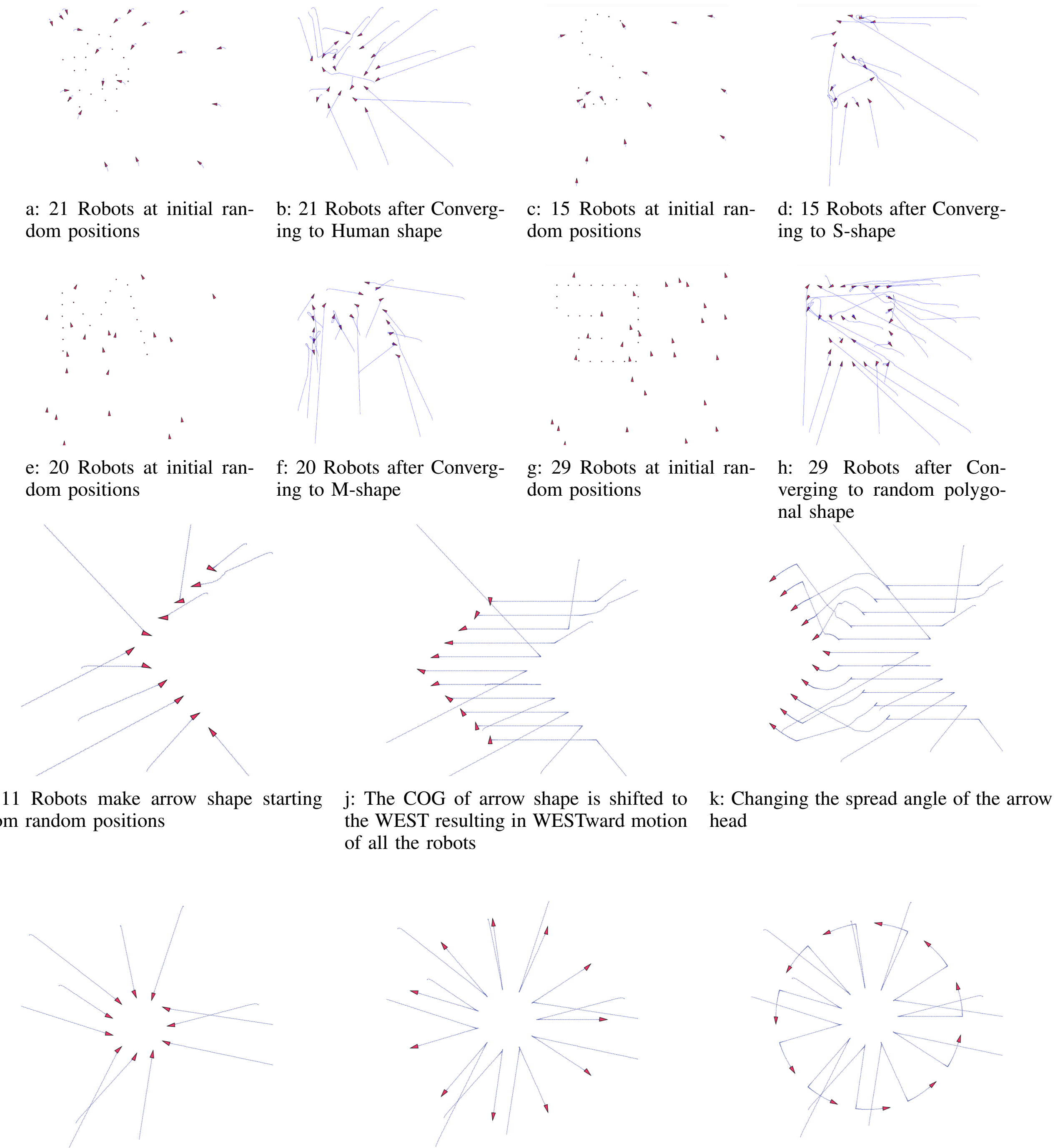
Here we discuss the effectiveness of our proposed method for the purpose of swarm pattern formation using two simulation-based experiments which we have already described in II. The performance of the proposed method was assessed both qualitatively and quantitatively with some metrics.

A. Experiment 1

The input images to the experiment are provided in Figures 3(a-d). Two snapshots of the robots after initializing at random positions and after converging to the pattern for each image are provided in Figures 5(a-h). Each blue line indicates the path followed by each robot. The robots can attain the formation extracted from images successfully. More versatile formation can also be formed effectively through this method. Observing the path followed by each robot, it is seen that the followed path is not the optimal one.

B. Experiment 2

As shown in Figure 5(i), the robots starting from random initial positions, form an arrow shape with 11 robots. Then the swarm is moved westward in Figure 5(j) and then the



a: 21 Robots at initial random positions

b: 21 Robots after Converging to Human shape

c: 15 Robots at initial random positions

d: 15 Robots after Converging to S-shape

e: 20 Robots at initial random positions

f: 20 Robots after Converging to M-shape

g: 29 Robots at initial random positions

h: 29 Robots after Converging to random polygonal shape

i: 11 Robots make arrow shape starting from random positions

j: The COG of arrow shape is shifted to the WEST resulting in WESTward motion of all the robots

k: Changing the spread angle of the arrow head

l: 11 Robots make circle shape starting from random positions

m: Changing the radius of the arrow shape, resulting in outward motion of the robots

n: Changing the heading angle of the swarm pattern resulting in anti-clockwise motion of the robots

Fig. 5: Snapshots of simulation of (a-h) experiment 1 where robots converge to swarm formation generated from 2D binary image and (i-n) experiment 2 where a swarm of robots try to form shapes generated with minimum shape parameters.

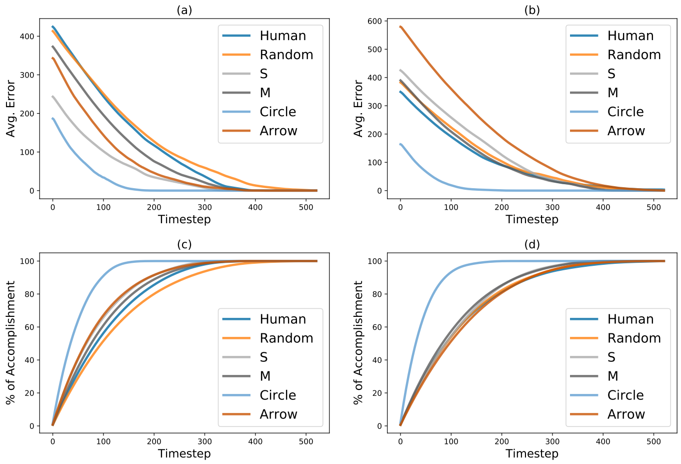


Fig. 6: The aggregated average error calculated over time-steps in 20 simulation runs for experiment 1 and 2 using (a) 20 robots and (b) 40 robots. The aggregated percentage of accomplishment is also plotted for the same experiments for (c) 20 robots and (d) 40 robots

spread angle is changed in Figure 5(k). The robots in Figure 5(l), again starts from initial random positions to form a circular shape. Thereafter, in Figure 5(m) and 5(n) the radius and heading angle are changed and the desired changes in formation are observed. From the figures, it can be observed that the robots can generate the desired shape robustly given the shape-specific parameters.

Running the same experiments 20 times for each shape the change of average error for that run was stored. In Figure 6(a-b) the average error for each robot was plot which was calculated using Eq. 12 with respect to simulation time step for various shapes formed by the swarm. In this plot we aggregated average error data from the simulation of pattern formation. For all the shapes, the robots fully settled to their target points in a time step range of $380-400$ for 20 robots and $409-440$ for 40 robots. From the plots, it is evident that the time required to converge is invariant of the shape complexity and changes with the increasing number of robots. The average percentage of target accomplishment as used in [13] was also plot in Figure 6(c-d) for 20 and 40 robots respectively. Other than a circle, for all the shapes the time of convergence almost remains the same and shifts forward in time for an increase in the number of robots. So the algorithm can effectively converge the swarm irrespective of shape complexity.

V. Conclusion

Two of the experiments showed the real-world applicability of our method and the time of convergence is found to be independent of the complexity of the shape. In addition, the use of a limited number of robot behaviors makes it easy to design and apply. In this paper, a hybrid, centralized, and application-oriented swarm pattern formation algorithm for non-holonomic robots by combining the idea of the virtual structure method for formation control with a behavior-based approach has introduced. The shapes to be formed by the robots can be extracted from images of object edges or can be designed using minimum control parameters. It is worth noting that not more than four parameters are required to compute the target position of each robot for the varied types of shapes that were demonstrated.

Although, the proposed method is more applicable in the real world than many other strategies, like all pattern formation strategies, this proposed strategy is not free from limitations. The average path followed by the robot is still not optimal and this method is not scalable.

Incorporating real-time applications can be considered as a future research direction of the proposed method. Calculating and improving movement efficiency will also be accounted for in our future works. The target point assigning method of each robot can be optimized by a population-based optimizer [14]. The network protocol to maintain the control hierarchy proposed in this method for ensuring robustness is yet to be designed. Lastly, the method is to be implemented on a real swarm of robots to confirm the results reported in the simulation of this paper.

References

[1] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics*. Springer, 2004, pp. 10–20.
[2] D. V. Dimarogonas and K. J. Kyriakopoulos, "Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 84–89.
[3] N. Y. Silvia, Husni, "The survey paper: Formation control for swarm robots," vol. 1, no. 58-62, Mar. 2016.
[4] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *mathematical Problems in Engineering*, vol. 2014, 2014.
[5] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous robots*, vol. 4, no. 4, pp. 387–403, 1997.
[6] J. Hu, A. Bruno, D. Zagieboylo, M. Zhao, B. Ritchken, B. Jackson, J. Y. Chae, F. Mertil, M. Espinosa, and C. Delimitrou, "To centralize or not to centralize: A tale of swarm coordination," *arXiv preprint arXiv:1805.01786*, 2018.
[7] N. Shiell and A. Vardy, "A bearing-only pattern formation algorithm for swarm robotics," 09 2016, pp. 3–14.
[8] J. Wiech, V. A. Eremeyev, and I. Giorgio, "Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following," *Continuum Mechanics and Thermodynamics*, vol. 30, no. 5, pp. 1091–1102, 2018.
[9] Z. Hendzel and J. Wiech, "Robotic swarm self-organisation control," *Acta Mechanica et Automatica*, vol. 13, no. 2, pp. 130–134, 2019.
[10] M. R. Pac, A. M. Erkmen, and I. Erkmen, "Control of robotic swarm behaviors based on smoothed particle hydrodynamics," oct 2007.
[11] Processing Foundation, "Processing." [Online]. Available: https://processing.org/
[12] C. W. Reynolds, "Steering behaviors for autonomous characters," pp. 763–782, 1999.
[13] H. Xu, H. Guan, A. Liang, and X. Yan, "A multi-robot pattern formation algorithm based on distributed swarm intelligence," vol. 1, pp. 71–75, 2010.
[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.