2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Decentralized Radial Segregation in Heterogeneous Swarms of Robots

Edson B. Ferreira Filho[1] and Luciano C. A. Pimenta[1]

***Abstract*— This article proposes a decentralized control strategy to reach radial segregation in heterogeneous robot swarms. The approach is based on a consensus algorithm applied to virtual points attached to each robot and a heuristics to compute the distance between the robots and the virtual point. Two scenarios are considered: when robots have access to a global reference point and when robots can communicate through a fixed underlying topology. A convergence proof is presented. Simulations and experimental results show that our approach allows a swarm of multiple heterogeneous robots to segregate radially using local information.**

I. INTRODUCTION

Recent interest in algorithms and applications for swarms of robots can attest the growth of this field. Some recent examples are an algorithm for swarms of robots that can fly and drive [1], algorithms and architectures for large swarms [2], [3] and a testbed for remote robotic experiments [4]. Some other examples deal with the use of heterogeneous swarms, as in [5] and in [6].

Swarms of robots are becoming more popular each year due to the perspective of having many simple robots solving real world complex problems with dexterity, scalability and fault tolerance. Some applications can make use of swarms composed of heterogeneous robots to increase the capabilities of the swarm. The heterogeneity of the swarm can be in the design of the robots or in their roles in the task to be performed.

There are many applications in which it might be interesting to have a heterogeneous swarm able to divide itself autonomously into groups containing only homogeneous robots. It is even more interesting if this division can be done using only local information in a decentralized manner.

Despite the great interest in robotic swarms, few works deal directly with the problem of segregating heterogeneous swarms of robots into clusters, among them we highlight [7], [8] and [9]. In [7] the problem of segregation for robots with double integrator dynamics is solved considering only two different groups and using potential functions. The work in [8] extends the ideas for multiple groups. In [9] a different approach is proposed using abstractions to represent each group and with the advantage that in some cases robots do not need global information. Even fewer works deal with the problem of segregating swarms of robots radially. A system is said to be radially segregated if all the robots of the same group are positioned at the same distance from a reference point while robots from different groups are positioned at different distances from a reference point. In the work of [10], robots of different groups have different dynamics and it seems that the system achieves radial segregation as a consequence instead of a goal of the approach. In [11] three mechanisms for sorting robots radially are presented briefly, in which two mechanisms are based on the fact that under centripetal forces particles segregate according to their size. In [12] the same fact is used to develop a controller to segregate robots radially with only one global information. The controller of [12] is then implemented in *e-puck* robots and shown in [13]. In all of these works, that tackle the problem of radial segregation, simulations are shown but no convergence proof is derived.

In this work, we propose a controller for robots with double integrator dynamics to radially segregate heterogeneous swarms. The robots know neither the number of robots nor the number of groups in the system. Different from other works we present a method with a proof of convergence. Thus, we can say that by using our controller the system will always reach a segregated state. Furthermore, our controller is local, thus with our controller robots do not need information from all the other robots of the system to achieve segregation as most of the works found in the literature.

II. BACKGROUND AND PRELIMINARIES

A. Problem Formulation

Consider N holonomic robots moving freely in a two dimensional Euclidean obstacle free environment. The dynamics of each robot is given by the double integrator

$$\dot{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{v}_{\boldsymbol{i}}, \quad \dot{\boldsymbol{v}}_{\boldsymbol{i}} = \boldsymbol{u}_{\boldsymbol{i}}, \quad i = 1, 2, ..., N; \qquad (1)$$

in which the position vector of each robot is given by $\boldsymbol{q}_{\boldsymbol{i}} = [x_i; y_i]^T$, the velocity vector by $\boldsymbol{v}_{\boldsymbol{i}} = [\dot{x}_i; \dot{y}_i]^T$ and the control input by $\boldsymbol{u}_{\boldsymbol{i}} = [u_{xi}; u_{yi}]^T$. Each robot is assigned to a group N_k, $k \in \mathcal{M} = \{1; 2; ..., M\}$ and M is the number of groups. Therefore, the system is composed of N robots divided into the groups $N_1, N_2, ..., N_M$. Robots of the same group are considered to be robots of the same type. Throughout this paper, indexes i and j are used to indicate the robots and indexes k and l are used to indicate groups.

Also, consider that each robot has a communication radius c that is the same for all the robots in the system. An example

This work was in part supported by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3, FAPESP (São Paulo Research Foundation) 2014/50851-0. This work was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (Finance Code 88887.136349/2017-00), and CNPq (grant number 311063/2017-9). Edson B. Ferreira Filho holds a scholarship from CAPES and Luciano C. A. Pimenta holds a scholarship from CNPq. Declarations of interest: none.

Edson B. Ferreira Filho and Luciano C. A. Pimenta (corresponding author) are with Graduate Program in Electrical Engineering - Universidade Federal de Minas Gerais - Belo Horizonte, MG, Brazil. emails: edsonbfilho@ufmg.br, lucpim@cpdee.ufmg.br.

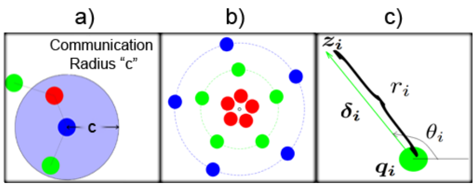

Fig. 1. Background and preliminaries Figs. a) Communication radius example. b) System of 15 robots divided in 3 groups radially segregated. c) Example of virtual point for a robot.

of this communication radius is showed in Fig. 1 (a), in which the bigger blue circle shows the radius c for the blue robot. The communication graph is built by considering the robots as nodes and defining edges between two robots if they are in the communication range of one another.

We aim to investigate the problem of radially segregating autonomous swarms of robots considering only the use of local information. All the robots should converge to a state where robots of the same type are positioned at the same distance with respect to a given point while these distances are different for robots of different types.

Fig. 1 (b) shows a system with 15 radially segregated robots divided in 3 groups in which robots of the same group have the same color. In Fig. 1 (b), dashed circles indicate that robots have indeed the same distance with respect to a given point that is represented by the small black circle.

In this paper we present a new approach to radially segregate swarms of robots. The approach is based on the control of virtual points associated with the robots. This idea of virtual points is better described next.

B. Virtual points

Consider that the *i-th* robot have a virtual point z_i associated with it. Fig. 1 (c) shows an arrow pointing to the virtual point of a robot. Each virtual point has an angle θ_i and a radius r_i associated with it. Those variables are the polar coordinates of the point as seen by a frame attached to the robot. We assume that each robot's virtual point is initialized with a random angle θ_i and the same distance $r_i = d$, in which d is a parameter dependent of c and will be defined in section III-B. Therefore, for the *i-th* robot of the system we have that $\boldsymbol{z_i} = \boldsymbol{q_i} + \boldsymbol{\delta_i}$, in which $\boldsymbol{\delta_i} = [r_i \cos\theta_i \quad r_i \sin\theta_i]^T$. The robot dynamics is given by the double integrator (1), thus to relate the motion of the robot with the motion of the virtual point we have that

$$\ddot{\boldsymbol{q}}_{\boldsymbol{i}} = \ddot{\boldsymbol{z}}_{\boldsymbol{i}} - \ddot{\boldsymbol{\delta}}_{\boldsymbol{i}}, \tag{2}$$

and differentiating $\boldsymbol{\delta_i}$ twice, with constant r_i, we have that

$$\ddot{\boldsymbol{\delta}}_{\boldsymbol{i}} = \begin{bmatrix} -r_i(\ddot{\theta}_i \sin\theta_i + \dot{\theta}_i^{\,2} \cos\theta_i) \\ r_i(\ddot{\theta}_i \cos\theta_i - \dot{\theta}_i^{\,2} \sin\theta_i) \end{bmatrix}. \tag{3}$$

In this differentiation we considered r_i as a constant. In our approach r_i will be allowed to change only at some specific discrete events instantaneously as it will be clear later when we present an algorithm to execute this change.

In order to achieve radial segregation in the system, we design a controller for the virtual points considering:

$$\ddot{\boldsymbol{z}}_{\boldsymbol{i}} = \hat{\boldsymbol{u}}_{\boldsymbol{i}}, \tag{4}$$

by enforcing $\boldsymbol{u_i} = \hat{\boldsymbol{u}}_{\boldsymbol{i}} - \ddot{\boldsymbol{\delta}}_{\boldsymbol{i}}$, from (1) and (2). This controller will be such that all virtual points converge to the same point eventually.

C. Required Information

Consider the group of all robots in the system: $\mathcal{R}$. Also consider the previously defined groups of robots of the same type N_k. We now define an ordered set of groups: $\mathcal{G} = \{N_1, N_2, ..., N_M\}$. We assume that this set of groups is a totally ordered set with a pre-defined binary relation ($<$). Consider the mapping that associates each robot to its corresponding group: $h : \mathcal{R} \rightarrow \mathcal{G}$. As $\mathcal{G}$ is a totally ordered set with a pre-defined binary relation, we can define a hierarchy such that:

$$h(R_{N1}) < h(R_{N2}) < ... < h(R_{NM}), \tag{5}$$

where R_{Nk} is any arbitrary robot of group N_k. We also define that $h(N_k)$ returns the corresponding value to the robots of group k. **We assume that robots do not have the information of how many groups there are in the system or how these robots are distributed in groups. Moreover, although we do not assume the robots know the whole set order, we do assume that they are able to compute the result of a comparison with robots of other groups according to the binary relation ($<$).** Thus, when robot i (R_i) meets robot j (R_j) they are able to access the result of the comparison $h(R_i) < h(R_j)$. This ability to compare will be useful when defining a radius heuristics to dynamically allocate different radius to different groups (Section III-B).

III. METHODOLOGY

In this work, the main idea consists in using a consensus based algorithm to drive virtual points attached to the robots and a heuristics to define the radius, r_i, of the virtual point. Concomitantly, robots spread themselves along the virtual circles where virtual points *rendezvous*.

We consider two different scenarios in which we use the same main idea for radial segregation. The scenarios differ mostly in the information which is available for the robots and the communication topology.

A. Consensus Algorithm

In both scenarios the radius heuristics (section III-B) and the angle controller (section III-C) are used in the same way. The only difference is in the consensus algorithm. In both scenarios we use a consensus algorithm as it is usually done in the context of the multi-robot *rendezvous* [14]. In this section we detail how the consensus controllers will differ from each other.

1) Scenario 1 - Underlying fixed communication topology: In this scenario, we consider that robots can communicate using a fixed underlying topology. Robots do not have the knowledge of a reference point. This topology must be connected and it is not dependent of the distance between pairs of robots. To control the virtual points we use the following consensus algorithm:

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}} = -\sum_{j=1}^{N} a_{ij}[(\boldsymbol{z_i} - \boldsymbol{z_j}) + \gamma(\dot{\boldsymbol{z}}_{\boldsymbol{i}} - \dot{\boldsymbol{z}}_{\boldsymbol{j}})], \quad (6)$$

in which γ is a positive gain, $\dot{\boldsymbol{z}}_{\boldsymbol{i}} = \dot{\boldsymbol{q}}_{\boldsymbol{i}} + \dot{\boldsymbol{\delta}}_{\boldsymbol{i}}$ and $a_{ij} = a_{ji}$ is given by the elements of an adjacency matrix from an arbitrary connected communication topology.

2) Scenario 2 - Robots know a reference point: In this scenario we consider that all the robots know exactly the location of a reference point. However, in this scenario robots cannot communicate outside a pre-espeficied range. We consider that the reference point is $\boldsymbol{o} = [0, 0]^T$, for the sake of simplicity.

We use the reference point in the consensus algorithm as if it were a fixed leader robot positioned at the reference point. Therefore, the global system configuration is given by: $\breve{\boldsymbol{q}} = [\boldsymbol{q_1}^T, \boldsymbol{q_2}^T, ..., \boldsymbol{q_N}^T, \boldsymbol{o}]^T$. Similarly, we consider also the extended $\breve{\boldsymbol{z}}$ and $\breve{\dot{\boldsymbol{z}}}$ (for $\boldsymbol{z}$ and $\dot{\boldsymbol{z}}$, respectively) in which we consider $\breve{\dot{\boldsymbol{z}}}_{N+1} = [0, 0]^T$ and $\breve{\boldsymbol{z}}_{N+1} = [0, 0]^T$. Thus, the consensus algorithm used for *Scenario 2* is:

$$\hat{\boldsymbol{u}}_{\boldsymbol{i}} = -\boldsymbol{z_i} - \gamma\dot{\boldsymbol{z}}_{\boldsymbol{i}} - \sum_{j=1}^{N} a_{ij}[(\boldsymbol{z_i} - \boldsymbol{z_j}) + \gamma(\dot{\boldsymbol{z}}_{\boldsymbol{i}} - \dot{\boldsymbol{z}}_{\boldsymbol{j}})], \quad (7)$$

in which a_{ij} is given by:

$$a_{ij} = a_{ji} = \begin{cases} 1, & \text{if} \quad ||\boldsymbol{q_i} - \boldsymbol{q_j}|| \leq c \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Note that when the virtual points converge to a *rendevouz* state, the distance from a robot to its virtual point will be the same as the distance from a robot to the reference point.

Note also that (6) and (7) are direct application of common consensus algorithms [15], [16]. In (6) and (7) we control virtual point positions to guide the virtual points (z) associated with all the robots to the same position.

Until now, nothing has been stated in regarding to the distances between the virtual points and the robots. In the next section we propose a heuristics to choose those distances for each robot dynamically to lead the system to radial segregation.

B. Radius Heuristics

To assign the virtual point radius r_i to each robot we propose a heuristics that dynamically changes robot's r_i when a robot is able to exchange information with other robots that are within the communication radius c. In *Scenario 1* robots exchange information through the underlying fixed communication topology (to be able to compute (6)) and also exchange information with other robots within the communication radius c (to be able to process Algorithm 1). In *Scenario 2* robots only exchange information with other robots within the communication radius c. The drawback for *Scenario 2* is that all the robots must know a reference point. This is not a strong limitation since we can always think of a two-stage solution in which in the first stage a preliminary consensus protocol might run while the robots stay still in their initial positions to define the reference point as long as robots start in a connected topology. The second stage is then exactly the proposed approach for *Scenario 2*. In Algorithm 1 we show the local control algorithm for robot i in which it is possible to see the heuristics to change the radius.

Algorithm 1: Control Algorithm for robot i.

Initialize : $h_i^d = 0$, $r_i = d$;
1 **while** *Active* **do**
2 Broadcast h_i, r_i, h_i^d;
3 **forall** $\mathbf{q_j}$ *such that* $\|\mathbf{q_j} - \mathbf{q_i}\| < c$ **do**
4 Receive h_j, r_j, h_j^d;
5 **if** $h_i > h_j$ **then** // Robot i is of a group higher in the hierarchy in comparison to robot j.
6 **if** $r_j + d > r_i$ **then**
7 $h_i^d \leftarrow 0$;
8 $r_i \leftarrow r_j + d$;
9 **if** $h_i = h_j$ **then** // Robots i and j are from the same group.
10 **if** $r_j + d > r_i$ **then**
11 $h_i^d \leftarrow 0$;
12 $r_i \leftarrow r_j$;
13 **if** $h_j > h_i$ *and* $r_j = r_i + d$ **then** // Robot i saving information.
14 $h_i^d \leftarrow h_i^d \cup \{(h_j, r_j)\}$;
15 **if** $\exists(h_k, r_k) \in h_j^d$ *such that* $h_i > h_k$ *and* $r_i \leq r_k$ **then** // Robot i analyzing received information.
16 $h_i^d \leftarrow 0$;
17 $r_i \leftarrow r_k + d$;
18 Move according to control law (13);

Each robot can perceive other robots within its communication radius and broadcast its own h_i, r_i and h_i^d (line 2). In Algorithm 1, h_i^d is used to store information about robots from other groups and then broadcast this information to other robots. The robots also receive the broadcasted h_j, r_j and h_j^d from all the other robots within its communication radius (line 4).

In Algorithm 1, lines 5-8, when a robot i meets a robot j of a group that is lower in the hierarchy than the group of robot i ($h_i > h_j$), with a radius that is greater or equal to r_i, robot i change its radius to r_j plus a fixed parameter d. This means that robot i, of the group higher in the hierarchy will move away from the "rendezvous point" thus segregating from the robot j, of the group lower in the hierarchy.

In Algorithm 1, lines 9-12, when a robot i meets a robot j from the same group, robot i receives the value of the radius of robot j if this value is greater than the one robot i already has. This means that robot j had met another robot from another group that is lower in the hierarchy and is now broadcasting this information to robot i.

Due to the local nature of the approach, there can be situations where robots are "stuck" in the same radius with robots of groups with lower order. These situations occurs when robots converged to the "wrong" circle but do not have communication with other robots in this circle. To solve these situations, whenever a robot i communicates with a robot j of a group higher in the hierarchy that is more external in relation to the reference point, robot i stores robot's j hierarchy (h_j) and radius (r_j) in a list (lines 13-14).

When a robot i receives the broadcasted list from robot j, robot i analyzes the list to see if robot j has information of a robot k that is from a group lower in the hierarchy than robot's i group ($h_i > h_k$). If yes and the radius of robot k is greater or equal than the radius of robot i ($r_k \geq r_i$) then robot's i radius should increase (lines 15-17). Informally, it works like one robot told the other: I have seen a robot with a group hierarchy smaller than yours with a radius greater or equal to the radius that you have, then you should increase your radius. This exchange of information can be better visualized in the video presented in `https://youtu.be/fohz_5DRmbI`.

To guarantee that the robots that are "stuck" have meetings with other robots, we make all robots rotate around the reference point with a polar angular velocity that depends on its radius r_i. Therefore, as robots rotate with different angular velocities, they will eventually meet other robots.

When robots are rotating, the robots from an external radius will eventually exchange information with robots of immediate internal radius as long as some conditions are met, as shown next. The constant d regulates the distances between groups and must be such that:

$$d < 0.5c. \tag{9}$$

Equation (9) guarantees that sometimes robots have connections with at least one robot of an internal group (if an internal group exists). Given the initial condition $r_i = d$, if an internal group does not exist, equation (9) guarantees that robots are able to communicate with at least one other robot of the system when reaching the circle of radius d. After updating its radius the robot moves according to the control law (13).

Now, to control the rotation of the robots around the reference point and to control the distribution of robots within the "desired radius" of its own group we define an angle controller as shown next.

C. Angle Control

It is preferable that robots of the same group distribute themselves uniformly along the virtual circle which is centered where the virtual points *rendezvous*, although it is not a requirement in the definition of the problem. Thus, we propose a controller for the dynamics of the angle θ_i, as follows

$$\ddot{\theta}_i = \tilde{u}_i = k_p\bar{\theta}_i + k_d\dot{\bar{\theta}}_i + k_\beta(\omega_i^d - \dot{\theta}_i), \tag{10}$$

in which k_p, k_d and k_β are positive gains. We also have that

$$\omega_i^d = k_\omega / r_i, \tag{11}$$

is the desired angle velocity, in which k_ω is a gain that regulates the fixed rotation for all robots of the system. Each robot will move locally to the mean angle in relation to its left and its right neighbors, as in [17]:

$$\bar{\theta}_i = \frac{{}^{Left}\theta_i + {}^{Right}\theta_i - 2\pi}{2} \tag{12}$$

in which ${}^{Left}\theta_i = \text{argmin}_{\tilde{\theta}_j \in \Omega'_j}\{\tilde{\theta}_j\}$ and ${}^{Right}\theta_i = \text{argmax}_{\tilde{\theta}_j \in \Omega'_j}\{\tilde{\theta}_j\}$. The set Ω is: $\Omega = \left\{\bigcup_{i=1}^N \theta_i\right\}$ and $\Omega'_i = \Omega \setminus \theta_i$ is the set containing the angles θ, of all robots of the same group of robot i, except the angle of robot i. We also have that $\tilde{\theta}_j$ is the measure of θ_j taken with respect to θ_i, i.e. $\tilde{\theta}_i = 0$.

The fixed angular velocity (ω_i^d) is always dependent on the robot radius r_i. The angular velocity is responsible for making robots eventually meet other robots if (9) is respected.

D. Control Law

We have proposed controllers for three different dynamics:

1) Consensus algorithm to control virtual point positions and velocities (section III-A);
2) Radius heuristics to dynamically set different radius for different groups (section III-B);
3) Angle controller to distribute robots of the same group (section III-C).

By combining those controllers, we can now define the complete control that will guide the movement of each robot. First we use the definition of (10) and the heuristics (Algorithm 1) to completely define (3). Then we use the consensus algorithm ((6) or (7)) to control the dynamics (4). Finally, composing (4) and (3) we define (2) and we can move the robots given by the dynamics of (1). Thus, each robot will be guided by the control law

$$\boldsymbol{u_i} = \boldsymbol{\hat{u}_i} - \begin{bmatrix} -r_i(\tilde{u}_i \sin\theta_i + \dot{\theta}_i^{\,2} \cos\theta_i) \\ r_i(\tilde{u}_i \cos\theta_i - \dot{\theta}_i^{\,2} \sin\theta_i) \end{bmatrix}, \tag{13}$$

in which $\boldsymbol{\hat{u}_i}$ will be either given by (6) or (7) depending of the considered scenario.

E. Controller Analysis

Theorem 1: Assume the facts:

(i) Individual robots are governed by the dynamics in (1) with communication radius c and constant parameter d such that $d < 0.5c$;

(ii) Groups and a binary relation between groups are defined in such a way that a strictly totally ordered set of groups is induced;

(iii) Each robot i is able to compute if the order of its group is greater, equal, or less than the order of the group of any other robot j according to the pre-defined binary relation when the information about the group of robot j is made available.

Then, by applying the Algorithm 1 in the control of each individual robot, it is guaranteed that the multi-robot system will converge to a radial segregation state as defined in Section II-A.

Proof: From fact (iii) we can assume that Algorithm 1 can run as all the comparisons can be properly computed.

In Algorithm 1 the motion of the robots is governed by (13). From (1) and (2) it is clear that the virtual points are driven by (4). In (4), $\hat{\boldsymbol{u}}_{\boldsymbol{i}}$ is determined according to the well-known consensus protocols in (6) or (7) which leads to the *rendezvous* of the virtual points. Given this fact, in order to show radial segregation we need to show that the radius r_i of each robot i converges to a given value which is the same value of the radius for robots of the same group and is a different value when compared to the radius of robots of other groups.

The rest of the proof employs induction, we first show that the first group will converge to the circle centered at the *rendezvous* point with radius given by d. Then, we follow to show that the other groups will converge to circles also centered at the *rendezvous* point but with radius that increases accordingly to its group order in the group hierarchy.

According to Algorithm 1, all the robots start with $r_i = d$ and the only possible changes in the radius implies that $r_i = \lambda d$, where $\lambda \in \mathbb{N}^*$. The changes can only occur when robots meet within a radius c, which is the same for every robot. Moreover, a radius never decreases, it might only increase in case robot i receives the information about the existence of another robot j of a different group so that $h_i > h_j$ and $r_i \leq r_j$ or another robot j of the same group so that $r_j > r_i$. As the set of groups is a strictly totally ordered set, and the changes are given by $r_i = r_j + d$ for $h_i > h_j$ or $r_i = r_j$ for $h_i = h_j$ it is guaranteed that the radius of the robots of the group which is the least element of the set, i.e. $h_1 < h_j \; \forall j$, never changes, i.e., $r_i = d$ which implies in the convergence to the circle centered at the *rendezvous* point with radius given by d.

Now consider the hypothesis: all the robots of group $1, 2, ..., k$, where $h(N_1) < h(N_2) < ... < h(N_k)$, have converged to the corresponding radius $r_{N1} = d, r_{N2} = 2d, ..., r_{Nk} = kd$. According to Algorithm 1 and the strict total order it is impossible to have a change in the radius of a robot of group N_{k+1} when meeting robots of groups $N_{k+2}, N_{k+3}, ..., N_M$ as $h(N_{k+1}) < h(N_{k+2}) < h(N_{k+3}) < ... < h(N_M)$. From this and the initial conditions, $r_i = d \; \forall i$, we can conclude that the radius of group N_{k+1} must converge to $r_{k+1} = \lambda d$ where $\lambda \in \{1, 2, 3, ..., k+1\}$.

We have that $d < 0.5c$ and the desired polar angular velocity ω_i^d given by (11) is so that robots moving at circles with different radius will move with different angular velocities. Thus, it is guaranteed that robots at the circle with $r_i = \lambda d$ receive information from the other robots at consecutive circles, i.e. $r_j = (\lambda + 1)d$ as they meet and share information periodically in finite time while they move in these circles. Moreover, in the first circle the robots have also access to the information from robots at the same circle. Thus, given the scheme of storing and broadcasting information of robots at consecutive circles in Algorithm 1 and given also the hypothesis of convergence of groups $N_1, ..., N_k$ we can guarantee that a robot from group N_{k+1} will always receive information about the existence of other robots at the same circle and the corresponding value $h(N_l)$ for comparison when $r_i = \lambda d$ with $\lambda \in \{1, 2, ..., k+1\}$. From this we can conclude that it is impossible for a robot of group N_{k+1} to converge to any circle of radius $r_i = \lambda d$ with $\lambda \in 1, ..., k$. According to Algorithm 1, being aware of the other robots already in their correct circle implies in the increment of the radius of the robots of group N_{k+1}. Therefore, the only possible circle for convergence is the one with $r_i = (k+1)d$.

By induction we can conclude that each robot i of group N_l will converge to $r_i = ld, \forall i, \forall l$. Thus, segregation will always be achieved. ■

We also have proposed an angle controller to distribute robots of the same group uniformly. As the angle controller only changes robots angles based on the proximity to other robots and virtual points remain unaltered, it does not interfere with the proof of $Theorem$ 1.

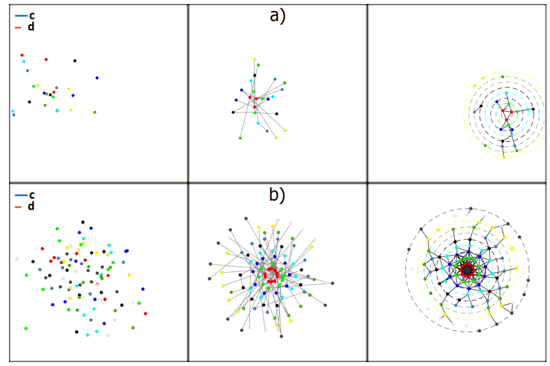

Fig. 2. Simulations in MATLAB. From top to bottom: (a) $N = 24$, $M = 8$. (b) $N = 100$, $M = 10$. From left to right, snapshots of initial to final iterations. In the middle snapshot of each simulation we highlight the fixed underlying topology. In the last snapshot we highlight in dashed lines the circles of the groups after segregation is reached and we also "connect" with black lines every robot that are within the communication radius in that instant.

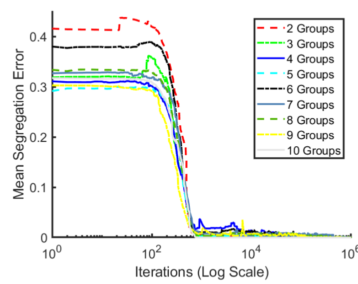

Fig. 3. Mean segregation error of 90 simulations with a varying number of robots and groups for *Scenario 2*.

IV. SIMULATIONS AND EXPERIMENTS

In this section we present two simulation runs for *Scenario 1* to analyze the proposal in a qualitative manner. For *Scenario 2* we present one of our trials of an experiment with real robots (using the *Robotarium* platform [4]) and we

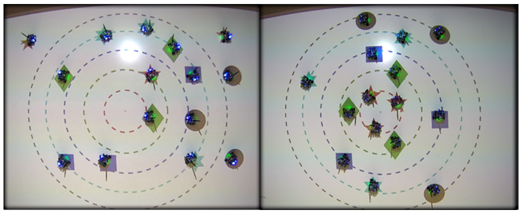

Fig. 4. Experiment in *Robotarium* [4]. Robots of the same group have the same color and form markers. Dashed circles represent the virtual circles for each group. Left: initial arbitrary positioned robots. Right: robots are radially segregated.

also present data of 90 simulations to evaluate the proposal in a quantitative manner.

A video with the simulations and the experiment can be found at: `https://youtu.be/yLZyN9MpC18`. In this video we show the evolution of the mean of information needed for each robot and the evolution of the segregation error (according to [12]) and the uniformity error (according to [11]) related to the simulations of Fig. 2. In the video we also show the experiment with real robots along with the evolution of the segregation error of the experiment. In all simulations we assume that all robots start with zero velocity and the robots were positioned according to a normal distribution. In both *Scenarios* the parameters were: $c = 5m$, $d = 0.5c - \epsilon$, $\epsilon = 0.1$, $\gamma = 5$, $k_p = 0.01$, $k_d = 0.01$, $k_\omega = 0.1$, $k_\beta = 10$ in which ϵ is set to guarantee (9).

In Fig. 2 we show two simulations for *Scenario 1*, one simulation with 24 robots and the other one with 100 robots. The underlying topology was arbitrarily set and is fixed and connected in each simulation.

In *Scenario 2* we performed 90 simulations varying the number of robots and groups ranging from two robots to 100 robots divided in two to 10 groups. Starting with two groups and one robot per group up to 10 robots and then increasing the number of groups: $3, 4, ..., 10$ groups. In Fig. 3 each line shows the mean segregation error for 10 simulations, from one to 10 robots per group and the number of groups is depicted in the legend.

In Fig. 4 we show an experiment with *15 GRITSBot X* [4] divided equally into 5 groups. The experiment was conducted for *236s* and segregation state was first reached around *100s* after the start. Parameters of the experiment: $c = 0.48$, $d = 0.5c - \epsilon$, $\epsilon = 0.05$, $\gamma = 8$, $k_p = 0.001$, $k_d = 0.005$, $k_\omega = 0.03$, $k_\beta = 0.1$. A local collision avoidance algorithm already implemented in the *Robotarium* platform [4] was used. After analyzing Figs. 2, 3 and 4 we can see that in all cases robots have reached radial segregation. It can also be seen in Fig. 2 that robots do not need information from all robots of the system to reach segregation. In average robots needed information of 10.43% and 24.20% of robots in the system, for the simulation in Figs. 2(a) and 2(b), respectively.

Note in Fig. 2 that robots can sometimes be unevenly distributed on the virtual circle of its group due to the local nature of the approach. Depending on the system's initial conditions some robots may never meet robots from the same group and consequently never reach perfect distribution. Nonetheless, they always reach the desired radius of their group. This would still be a radially segregated system according to our definition of the problem.

V. CONCLUSIONS

In this paper we have presented a novel decentralized approach to radially segregate swarms of heterogeneous robots. We have shown proof of convergence for two different scenarios: when robots have an underlying communication topology and when robots have the knowledge of a common reference point. In our approach robots do not need information about all the robots of the system, as was the case in some previous works. Future work will focus on collision avoidance strategies integrated with our controller.

REFERENCES

[1] Araki, B., Strang, J., Pohorecky, S., Qiu, C., Naegeli, T., and Rus, D. (2017). Multi-robot path planning for a swarm of robots that can both fly and drive. *IEEE Int. Conf. on Robotics and Automation, pp. 5575-5582.*

[2] Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems, 62(7), 966-975.*

[3] Preiss, J. A., Honig, W., Sukhatme, G. S., and Ayanian, N. (2017). Crazyswarm: A large nano-quadcopter swarm. *IEEE Int. Conf. on Robotics and Automation, pp. 3299-3304.*

[4] Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M. (2017). The Robotarium: A remotely accessible swarm robotics research testbed. *IEEE Int. Conf. on Robotics and Automation, pp. 1699-1706.*

[5] Gansari, M., and Buiu, C. (2017). Software system integration of heterogeneous swarms of robots. *Journal of Control Engineering and Applied Informatics, 19(3), 49-57.*

[6] Espinosa, G., and Rubenstein, M. (2018). Using Hardware Specialization and Hierarchy to Simplify Robotic Swarms. *IEEE Int. Conf. on Robotics and Automation, pp. 7667-7673.*

[7] Kumar, M., Garg, D. P., and Kumar, V. (2010). Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Trans. on Automatic Control, 55(3), 743-748.*

[8] Santos, V. G., Pimenta, L. C., and Chaimowicz, L. (2014). Segregation of multiple heterogeneous units in a robotic swarm. *IEEE Int. Conf. on Robotics and Automation, pp. 1112-1117.*

[9] Ferreira Filho, E., and Pimenta, L. (2015). Segregating multiple groups of heterogeneous units in robot swarms using abstractions. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 18(1): 121-129.*

[10] Szwaykowska, K., Romero, L. M. Y. T., and Schwartz, I. B. (2015). Collective motions of heterogeneous swarms. *IEEE Trans. on Automation Science and Engineering, 12(3), 810-818.*

[11] Wilson, M., Melhuish, C., Sendova-Franks, A. B., and Scholes, S. (2004). Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots, 17(2-3), 115-136.*

[12] Groß, R., Magnenat, S., and Mondada, F. (2009). Segregation in swarms of mobile robots based on the brazil nut effect. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 4349-4356.*

[13] Chen, J., Gauci, M., Price, M. J., and Groß, R. (2012). Segregation in swarms of e-puck robots based on the brazil nut effect. *Int. Conf. on Autonomous Agents and Multiagent Systems. vol. 1: 163-170.*

[14] Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on automatic control, 48(6), 988-1001.*

[15] Ren, W. and Atkins, E. (2007). Distributed multivehicle coordinated control via local information exchange. *Int. Journal of Robust and Nonlinear Control: IFAC - Affiliated Journal, 17(1011), 1002-1033.*

[16] Ren, W. (2007). Second-order consensus algorithm with extensions to switching topologies and reference models. *In American Control Conf., pp. 1431-1436.*

[17] Gonçalves, M., Pimenta, L., and Pereira, G. (2011). Coverage of curves in 3D with swarms of nonholonomic aerial robots. *18th IFAC World Congress, pp. 10367-10372.*