2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Segregating Multiple Groups of Heterogeneous Units in Robot Swarms using Abstractions

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

Abstract— **This paper addresses the problem of segregation of groups of heterogeneous units in robot swarms. We propose a controller that can drive robots in a way that each group composed of robots of a similar type will form clusters while maintaining segregation from other groups. The approach is based on abstractions created to represent each group of robots and an artificial potential function used to segregate the groups. Different from previous works on swarm segregation, we can mathematically guarantee that by using our approach the system will always converge to a state where multiple dissimilar groups are segregated. Moreover, in some situations, our controller does not require all robots to have information about all the other robots in the system. We demonstrate the effectiveness of our controller with simulations with different types of robots and varying number of robots and groups.**

I. INTRODUCTION

Swarms of robots are systems composed of a number of autonomous agents that need to interact and cooperate to achieve a common goal [1]. These systems are characterized by decentralized control, limited communication between robots, use of local information, and emergence of global behavior [2]. The first researcher to reproduce computationally the behavior of animal swarms was Reynolds [3], the goal was the automation of those behaviors in graphics computation. Since then, different works have addressed the problem of controlling swarms of robots [4], [5], [6], [7]. There are multiple advantages inherent to swarms such as fault tolerance due to the redundancy in its construction. Some recent works are now focusing on applications of swarms of robots, such as perimeter surveillance [8], spill detection [9], interactions with humans [10], [11] among others [12], [13].

Swarms of heterogeneous robots are those composed of different types of robots, either in its design or in its role in the task to be performed. For example, one can design a system of heterogeneous robots to be used in perimeter surveillance where some robots have cameras and are responsible ford the surveillance while some other robots are designed to warn humans if there is a breach in the perimeter.

Some works address the use of heterogeneous robots in different contexts. For example, Dorigo [2] proposed a scheme in which two different types of ground robots and one aerial robot work together and Pimenta [14] proposed the collaboration of robots with heterogeneous sensing capabilities.

In the case of heterogeneous swarms an important ability of the system which might be useful in several applications is the capacity of autonomous segregation. This is the ability of forming groups, each one composed solely of robots of the same kind. In order to provide this capacity to the system one must design individual control laws that make robots of the same group form clusters while maintaining distance from other groups.

Few works directly tackle the segregation problem. Studies relevant in the solution of the segregation problems are [15], [16], [17] and [18].

Groß [15] developed a centralized algorithm that can segregate robots based on the *Brazilian nut effect*, in which nuts are segregated based on its granulometry. This algorithm is then implemented [17] in *e-puck* robots.

Kumar et al. [16] used an artificial potential function based in the *differential adhesion model* for biological cells. They showed a proof of asymptotic convergence to segregation and stability analysis of a robotic swarm with only two groups.

Kumar's work [16] was then extended in [18] with a similar potential function that is capable of segregating more than two groups. Santos's [18] and Kumar's [16] controllers are distributed, although in both cases each robot needs information from every other robot in the system during all the time.

This paper presents a controller that differs from previous works [16], [18]. The controller is based on the use of abstractions [4] to represent each group of robots and an artificial potential function [19] to create the artificial force that segregate the groups represented by the abstractions.

Our controller has two clear advantages. The first one is with respect to the convergence to segregation with multiple groups. In [16], it is shown the convergence to segregation with only two groups of heterogeneous robots and in [18] it is only shown stability but not convergence for multiple groups. In this work we show convergence to segregation with multiple groups of heterogeneous robots. The second advantage is that our controller might not require that each robot receive information from all the other robots in the system, during all the time. This property appears particularly when there is a great number of groups in the system, in that case, each robot will only need information from the robots of its own group and from groups called neighbors most of the time.

This paper is organized in five sections. Section II presents

*This work has been supported by the Brazilian agency CAPES. This work has also been supported by the Brazilian agencies CNPq and FAPEMIG.

[1]Edson Bernardes Ferreira Filho and Luciano Cunha de Araújo Pimenta are with Graduate Program in Electrical Engineering - Federal University of Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil. emails: edsonbffilho@ufmg.br, lucpim@cpdee.ufmg.br

978-1-4799-9993-4/15/$31.00 ©2015 IEEE

the segregation problem formulation. Section III is composed of four parts that leads to the proposed controller and its convergence proof. In section IV, a simulation is shown and the results are discussed. Section V concludes this paper with final observations and future work perspective.

II. PROBLEM FORMULATION

Consider N holonomic robots moving freely in a two dimensional Euclidean space, as in the other works [18] and [19]. The dynamics of each robot is given by the double integrator

$$\dot{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{v}_{\boldsymbol{i}}, \quad \dot{\boldsymbol{v}}_{\boldsymbol{i}} = \boldsymbol{u}_{\boldsymbol{i}} \quad i = 1, 2, ...N, \tag{1}$$

where 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_j, $j \in \mathcal{M} = \{1, 2, ...M\}$ 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.

In this paper, we are interested in the segregation problem [16], [18]. This is the problem of designing a control law that drives the system to a state in which robots of the same type or group form clusters separated from the robots of other types. When this state is reached, the system is said to be segregated. In this work we will assume that each group of robots of the same type is represented by an abstraction [4], which is invariant to robots permutations and with dimension independent of the number of robots. More specifically, each abstraction ϕ_j, $j \in 1, ..., M$ will be defined by a circle with mean μ_j and radius R_j. Now, we can formally state our version of the segregation problem to be solved:

Problem Statement 1: Given N robots with dynamics given by (1) of M types, where $N \geq M$, design individual control laws u_i that guarantee that each robot i remains in the interior of the abstraction ϕ_j that represents the robots of the same type of robot i and at the same time each abstraction ϕ_j converges to a state where:

$$\bigcap_{j=\{1,...,M\}} \phi_j = \varnothing. \tag{2}$$

Figure 1 shows a system segregated according to our definition.

III. METHODOLOGY

We propose a strategy to guarantee the segregative behavior of the system. This strategy consists mainly in coupling two ideas: the use of abstractions to represent each group and an artificial potential function to segregate those groups. Then we encapsulate both ideas into a control law which is used by each robot.

A. Abstractions

The abstraction considered in this work is the same one defined in [4], in the context of motion planning for large multi-robot systems. Each abstraction is defined using the

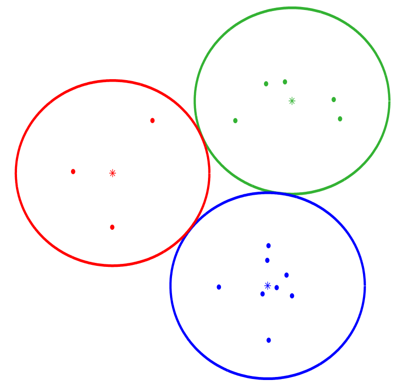

Fig. 1. $N = 16$ point robots unevenly distributed into $M = 3$ groups. Robots of the same type have the same color and are inside the same abstraction. Asterisks represent abstractions center (μ_j).

mean and covariance of the positions of all robots in a group. The mean of each group is given by

$$\boldsymbol{\mu}_{\boldsymbol{j}} = \begin{bmatrix} \mu_j^x \\ \mu_j^y \end{bmatrix} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{q}_{\boldsymbol{i}}, \tag{3}$$

where n is the number of robots in the group associated with the abstraction ϕ_j. In the two dimensional Euclidean space, physically, the abstraction is a circle with center given by (3) and "size" given by

$$\sigma_j = \frac{1}{n}\sum_{i=1}^{n}((x_i - \mu_j^x)^2 + (y_i - \mu_j^y)^2) = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{q}_{\boldsymbol{i}} - \boldsymbol{\mu}_{\boldsymbol{j}}\|^2 . \tag{4}$$

The radius of the abstraction circle is $R_j = \sqrt{n\sigma_j}$. As shown in [20] it is easy to see that by definition, the robots associated with ϕ_j remains inside the circle with R_j. Note that $\|\boldsymbol{q}_{\boldsymbol{i}} - \boldsymbol{\mu}_{\boldsymbol{j}}\|^2 \leq \sum_{i=1}^{n} \|\boldsymbol{q}_{\boldsymbol{i}} - \boldsymbol{\mu}_{\boldsymbol{j}}\|^2 = n\sigma_j \Rightarrow \|\boldsymbol{q}_{\boldsymbol{i}} - \boldsymbol{\mu}_{\boldsymbol{j}}\| \leq \sqrt{n\sigma_j}$. It is important to mention that if $n = 1$, then the circle is degenerated to a point given by the position vector $\boldsymbol{q}_{\boldsymbol{i}}$. Formally, the abstraction ϕ_j is a surjective submersion $\phi_j = \mathbb{R}^{2n} \rightarrow \mathbb{R}^3$ mapping from the original configuration space to a lower dimensional space:

$$\phi_j(\hat{\boldsymbol{q}}_{\boldsymbol{j}}) = \begin{bmatrix} \mu_j^x & \mu_j^y & \sigma_j \end{bmatrix}^T, \tag{5}$$

in which $\hat{\boldsymbol{q}}_{\boldsymbol{j}}$ is a vector composed of the position of all the robots of a given group: $\hat{\boldsymbol{q}}_{\boldsymbol{j}} = [x_1, y_1, x_2, y_2, ..., x_n, y_n]^T$.

In order to design our individual controllers it is important to relate the motion of the abstraction with the motion of the robots. Thus, differentiating (5)

$$\dot{\phi}_j = d\phi_j \dot{\hat{\boldsymbol{q}}}_{\boldsymbol{j}}. \tag{6}$$

By using (3), (4) and (5) we can obtain $d\phi_j$:

$$d\phi_j = \frac{1}{n}\begin{bmatrix} 1 & 0 & 2(x_1 - \mu_j^x) \\ 0 & 1 & 2(y_1 - \mu_j^y) \\ \vdots & \vdots & \vdots \\ 1 & 0 & 2(x_n - \mu_j^x) \\ 0 & 1 & 2(y_n - \mu_j^y) \end{bmatrix}^T . \tag{7}$$

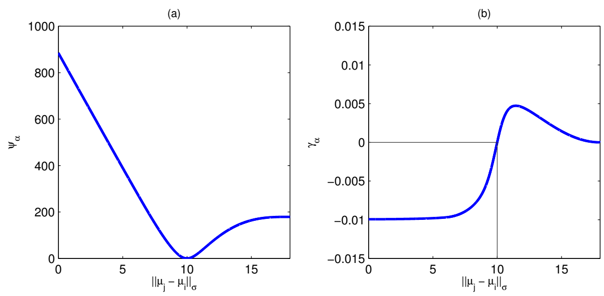

Fig. 2. Parameters: $h = 0.3$, $c = 0.01$, $d_\alpha = 10$ and $r_\alpha = 1.8d_\alpha = 18$ (a) Example of artificial potential function interaction between two agents versus the distance between them. (b) Gradient based force between two agents versus the distance between them.

Since we have agents with double integrator dynamics, we need a relation between the abstraction motion and the robots acceleration. By differentiating (4) twice we have

$$\ddot{\sigma}_j = \frac{2}{n}\begin{bmatrix} x_1 - \mu_j^x \\ y_1 - \mu_j^y \\ x_2 - \mu_j^x \\ y_2 - \mu_j^y \\ \vdots \\ x_n - \mu_j^x \\ y_n - \mu_j^y \end{bmatrix}^T \ddot{q}_j + 2\sigma_j', \tag{8}$$

where,

$$\sigma_j' = \frac{1}{n}\sum_{i=1}^{n}(\dot{x}_i - \dot{\mu}_j^x)^2 + (\dot{y}_i - \dot{\mu}_j^y)^2. \tag{9}$$

Now, we can write:

$$\ddot{\phi}_j = d\phi_j \ddot{q}_j + \begin{bmatrix} 0 \\ 0 \\ 2\sigma_j' \end{bmatrix}. \tag{10}$$

In section III-C we are going to propose a control law for each robot of the group so that the corresponding abstraction dynamics in (10) can be simplified to:

$$\ddot{\phi}_j = w_j, \tag{11}$$

where w_j is a virtual input for the abstraction which will be given by:

$$w_j = \begin{bmatrix} k_\mu \boldsymbol{U_j^\mu} & U_j^\sigma \end{bmatrix}^T, \tag{12}$$

where $\boldsymbol{U_j^\mu}$ is an artificial force that guides the motion of the group mean, U_j^σ determines the evolution of the abstraction size and k_μ is a positive gain. Our choice of $\boldsymbol{U_j^\mu}$ and U_j^σ will determine the success of our strategy.

B. Potential function

This section describes an artificial potential function with a *finite cutoff* inspired by the one first presented in [19] with the aim of generating a proper artificial force $\boldsymbol{U_j^\mu}$.

Before showing the potential function we need to define the σ-norm of a vector. This is a map $\mathbb{R}^m \to \mathbb{R}_{\geq 0}$ given by [19]:

$$\|z\|_\sigma = \frac{1}{\epsilon}[\sqrt{1 + \epsilon\|z\|^2} - 1], \tag{13}$$

where ϵ is a parameter larger than zero to guarantee that $\|z\|_\sigma$ is differentiable everywhere. For the sake of simplicity, in this work we use $\epsilon = 1$.

Now, considering the means of the abstractions as the high level agents to be guided, the collective potential function defined in [19] is given by:

$$V(\mu) = \frac{1}{2}\sum_i \sum_{j \neq i} \psi_\alpha(\|\boldsymbol{\mu_j} - \boldsymbol{\mu_i}\|_\sigma), \tag{14}$$

where

$$\psi_\alpha(z) = \int_{d_\alpha}^{z} \gamma_\alpha(s)ds, \tag{15}$$

$$\gamma_\alpha = \rho_h(z/r_\alpha)\frac{c(z - d_\alpha)}{\sqrt{1 + (z - d_\alpha)^2}}, \tag{16}$$

and function $\rho_h(z)$ is a *bump* function, that smoothly varies from 1 to 0:

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2}\left[1 + cos(\pi(\frac{z-h}{1-h}))\right] & z \in [h, 1] \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Figures 2(a) and 2(b) show examples of functions ψ_α and γ_α respectively. Parameter c is a gain of the function, and parameter h acts in the smoothness of the corresponding gradient. The parameters r_α and d_α are the finite cut-off $r_\alpha = \|r\|_\sigma$ and the global minimum of ψ_α, $d_\alpha = \|d\|_\sigma$, respectively (see Figure 2).

By using the artificial potential function previously shown, we can define artificial forces composed of two terms:

$$\boldsymbol{F_i} = \sum_{j \in B_i} \gamma_\alpha(\|\boldsymbol{\mu_{ij}}\|_\sigma)\boldsymbol{n_{ij}} + \sum_{j \in B_i} \rho_h(\|\boldsymbol{\mu_{ij}}\|_\sigma / r_\alpha)(\dot{\boldsymbol{\mu}}_{\boldsymbol{j}} - \dot{\boldsymbol{\mu}}_{\boldsymbol{i}}), \tag{18}$$

where $\boldsymbol{n_{ij}}$ is a vector pointing in the direction $\boldsymbol{\mu_j} - \boldsymbol{\mu_i}$:

$$\boldsymbol{n_{ij}} = \frac{(\boldsymbol{\mu_j} - \boldsymbol{\mu_i})}{\sqrt{(1 + \|\boldsymbol{\mu_{ij}}\|^2)}}, \tag{19}$$

where $\boldsymbol{\mu_{ij}} = \boldsymbol{\mu_i} - \boldsymbol{\mu_j}$ and B_i is the neighborhood of group i, i.e. these are the other groups so that the distance $\|\boldsymbol{\mu_i} - \boldsymbol{\mu_j}\|$ is less than r, which is the parameter that defines the finite cut-off of the potential function.

First term of (18) is a gradient term based on (14), and the second term acts as a *velocity damping*, where $\dot{\boldsymbol{\mu}}_{\boldsymbol{i}}$ and $\dot{\boldsymbol{\mu}}_{\boldsymbol{j}}$ are velocities of the centers of abstractions i and j, respectively.

In the case where the parameters r and d are such that $d < r < 2d$, the important *Lemma*, which is proved in [19], holds:

Lemma 1: (Lemma 3 in [19]) Every local minima of $V(\mu)$ is an $\alpha-$*lattice* and vice-versa.

An $\alpha-$*lattice* is a formation such that the following set of algebraic constraints hold:

$$\|\boldsymbol{\mu_j} - \boldsymbol{\mu_i}\| = d, \forall j \in B_i. \tag{20}$$

This Lemma will be useful in the proof of convergence of the proposed controller in the next section.

C. Control law

We can now show the individual control laws:

$$\boldsymbol{u_i} = d\phi_j^T (d\phi_j d\phi_j^T)^{-1} \left\{ - \begin{bmatrix} 0 \\ 0 \\ 2\sigma'_j \end{bmatrix} + w_j \right\}, \quad (21)$$

where ϕ_j is the abstraction of robot i and w_j has to be designed to control the state of the abstraction. Note that $det(d\phi_j d\phi_j^T) = \frac{(2\sigma_j)}{n^3}$, then as long as $\sigma_j \neq 0$, the determinant is different from zero which means that the inverse always exist. From (10) by applying this control law in every robot, each abstraction will move according to w_j, as follows:

$$\ddot{\phi}_j = w_j = \begin{bmatrix} k_\mu \boldsymbol{U_j^\mu} \\ U_j^\sigma \end{bmatrix}. \quad (22)$$

We design w_j by using two components, $\boldsymbol{U_j^\mu}$ and U_j^σ. Component $\boldsymbol{U_j^\mu}$ guides the motion of the mean of abstraction j and it is defined by the artificial forces in (18):

$$\boldsymbol{U_j^\mu} = \boldsymbol{F_j}. \quad (23)$$

This choice of artificial force will guide the system to form an $\alpha-$lattice with parameter d.

We design component U_j^σ in order to reach the desired size for each abstraction. To guarantee that the segregation will be obtained according to our definition in (2), we have to specify the proper desired size of each abstraction. This desired size has to be specified so that the distance between the means of two abstractions will be greater than the sum of the radius of both abstractions when the system reaches convergence, a suficient condition is that, the radius R_j of each abstraction is less than half the distance d in the $\alpha-$lattice. We know that the radius of each abstraction is $R_j = \sqrt{n\sigma_j}$ [20], and we now impose that $R_j < (d/2)$, to guarantee segregation as $t \to \infty$. Thus, we define the parameters so that:

$$\sqrt{n\sigma_j^{des}} < \frac{d}{2}, \quad (24)$$

or

$$\sigma_j^{des} < \frac{(d^2)}{4n}, \quad (25)$$

where σ_j^{des} is the desired value for σ_j.

Now, we propose the following dynamics for U_j^σ:

$$U_j^\sigma = \ddot{\sigma}_j^{des} + k_1(\dot{\sigma}_j^{des} - \dot{\sigma}_j) + k_2(\sigma_j^{des} - \sigma_j), \quad (26)$$

where k_1 and k_2 are properly designed positive gains, and

$$\dot{\sigma}_j = \frac{2}{n} \sum_{i=1}^{n} (x_j^i - \mu_j^x)\dot{x}_j^i + (y_j^i - \mu_j^y)\dot{y}_j^i. \quad (27)$$

We can set σ_j^{des} as a constant, $\ddot{\sigma}_j^{des}$ and $\dot{\sigma}_j^{des}$ to zero, so that the abstraction size will have zero velocity and zero acceleration when $t \to \infty$. Thus, each individual robot will be guided by the control law composed of (21), (23), (25) and (26), as follows:

$$\boldsymbol{u_i} = k_\mu \boldsymbol{U_j^\mu} + \frac{(\boldsymbol{q_i} - \boldsymbol{\mu_j})}{\sigma_j}[2\sigma'_j - k_1\dot{\sigma}_j + k_2(\frac{d^2}{4n} - \delta - \sigma_j)], \quad (28)$$

in which δ is a positive small value to guarantee (25). Each abstraction can have a different number of robots n. Parameters k_μ, k_1, k_2 and δ are fixed and equal to all abstractions.

Individual control law (28) is dependent on the number of robots in the abstraction, the state of the robot itself $(q_j, \dot{q}_j)$, the state of the abstraction ϕ_j and the state of the neighbor abstractions.

D. Controller analysis

In this section we formally analyze the proposed controller to demonstrate its effectiveness to solve the problem of segregation.

Theorem 1: Applying individual control law (28) in the system with M groups and N robots with dynamics given by (1), and assuming we do not have a situation where all robots of the same group are placed at the same position at the same time and the system does not start at a local maximum or saddle point of function $V(\mu)$ in (14), the system will converge to segregation i.e, the problem defined in the *Problem Statement 1* will be solved.

Proof: Our method was constructed to guarantee the solution of the problem, which means that our proof is straight forward. The analysis is conducted in two parts. First, we have to prove that all robots in an abstraction will stay inside it and the abstraction state will converge to the desired size. The second part is to show that the abstractions will end separated apart without intersections.

Given the assumption that the robots are not at the same position at the same time, the determinant of $d\phi_j d\phi_j^T$ is different from zero and the inverse in (21) always exist, then the motion of the abstraction will be given by (22). From (26) it should be clear that if k_1, k_2 are properly designed the dynamics given by $\ddot{\sigma}_j = U_j^\sigma$ will be such that σ_j converges to σ_j^{des} exponentially [21]. Since the radius is defined according to $R_j = \sqrt{n\sigma_j}$, we know from section III-A that the robots of ϕ_j will remain inside the abstraction during all the time.

For the second part of the proof, we consider the proof of *Theorem 1* in [19]. In this theorem, *LaSalle's invariance principle* is used to show that a set of agents with double integrator dynamics subject to the artificial potential force in (18) (see *Algorithm 1* in [19]) asymptotically converges to a configuration which is an equilibrium of function V. Since we assume that the system does not start at a local maximum or at a saddle point of V and these are unstable equilibria we can guarantee that the system asymptotically converges to a local minimum of V. By using *Lemma 1* (see III-B) we can conclude that the system asymptotically converges to an $\alpha-$lattice formation.

As the abstractions reach the desired size, with all the robots of the abstraction inside, together with the fact that the other parameters (see (25)) were specified to guarantee absence of intersections among abstractions when forming the $\alpha-$lattice, then the problem of segregation as defined in the *Problem Statement 1* will be solved as $t \to \infty$. ■

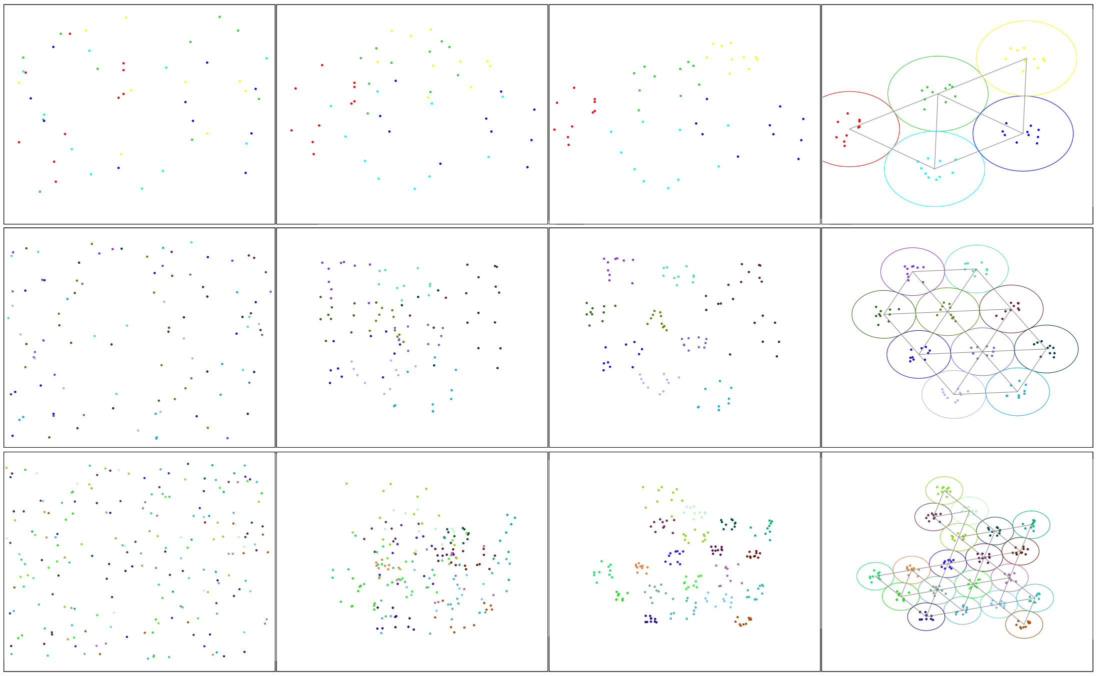

Fig. 3. Simulations in MATLAB, each group has $n = 10$ robots. From top to bottom: (a) $M = 5$ groups. (b) $M = 10$ groups. (c) $M = 20$ groups. From left to right, 4 snapshots of initial to final iterations. Last snapshot of each simulation also highlight the abstraction size and the formation of the $\alpha-$lattices.

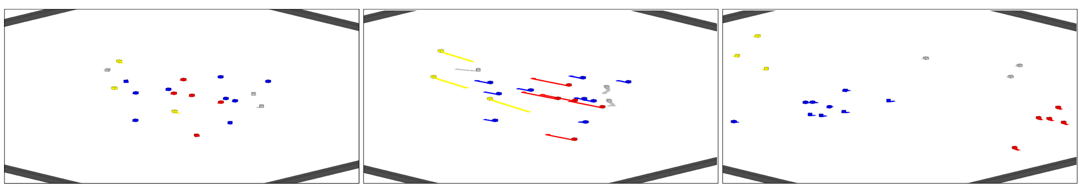

Fig. 4. Snapshots of simulation in ROS/Stage with 20 robots distributed unevenly in 4 groups. Each group is represented by a different color. From left to right we have the initial simulation step, a intermediary step and the final step.

IV. NUMERICAL EXPERIMENTS

We tested the proposed controller in two different platforms ROS/Stage and MATLAB. In ROS/Stage we used *differential drive* robots and in MATLAB we used a model of holonomic robots. In this section we present one simulation run in ROS/Stage and three runs in MATLAB. In addition, we discuss the results and the advantages and limitations of the method. A video of all four simulations is found on the web: http://youtu.be/mFxI3YQrhSk.

A. Simulations

In all simulations we assume that all robots start with zero velocity and the robots were positioned according to a normal distribution. The simulations were performed in two environments. In MATLAB, we performed simulations to test the feasibility of our approach with a varying number of robots and groups. In ROS/Stage we are interested in analyze our controller in a more realistic simulation environment.

We performed extensive simulation in MATLAB, using a holonomic robot model. Potential function parameters were assumed: $r = 1.5d$, $h = 0.1$, $c = 50$, $\delta = 0.01$. Gains k_μ, k_1 and k_2 were set to $10, 5, 50$ respectively. Other parameters, such as the desired distance between groups and the normal distribution of robots were set in a way that we can better visually evaluate our approach and are dependent on the number of groups and robots. The simulations were stopped as soon as segregation was reached according to our definition in (2). Simulations with 5, 10 and 20 groups of robots are shown in Figure 3.

In order to better depict the "local" property of our controller in comparison to the works in [16] and [18], Figure 5 shows the average number of groups in neighborhood B_i versus the iterations, that is, the average amount of information needed for robots from initial time to the time segregation was reached.

In ROS/Stage we performed simulations with a different purpose. Figure 4 shows an example of this simulations. We aim to show the applicability of our controller with differential drive robots in unbalanced groups. Groups are composed of 20 robots divided in 9, 5, 3 and 3 robots per group. We made use of the *feedback linearization* [6] approach in order to use the designed controller with the differential drive robot. The simulation was stopped when, visually, the robots were segregated.

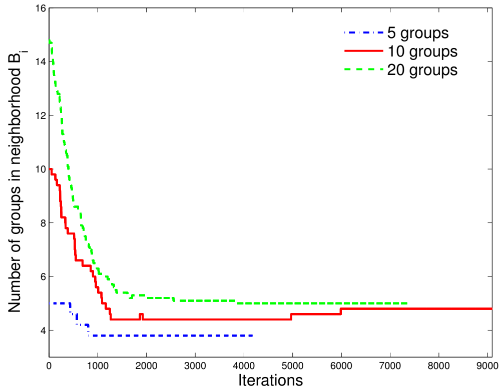

Fig. 5. Information of how many groups each robot needs (average) versus iterations. e.g. With 10 and 20 groups, after the iteration 2000, each robot needs information only of less than 6 neighbor abstractions.

B. Discussion

After the simulations it is easy to see that our approach is different from Kumar's and Santos's works. Our experiments showed segregative behavior independent of the total number of groups and robots and how robots are distributed into groups. The work in [18] verified problems in segregation in the case of unbalanced groups.

The main advantage in our approach is that we can formally guarantee that segregation will be always achieved when $t \to \infty$. Another advantage of our controller is that in many situations robots only need local information to segregate. As can be seen in Figure 5, as the clusters begin to be formed, the amount of information needed by each group decreases. Even with 10 and 20 groups, after some time, robots only need the states of the robots of its own group and the state of, in average, 5 other abstractions. In Figure 5 we can also see that if we increase the number of groups in the system, the amount of information needed by each group is not proportionally increased.

Our controller is also robust to adding or subtracting robots in the system as long as the condition in (25) is always satisfied.

A current limitation of our approach is the lack of a collision avoidance strategy. In practical situations, a low-level collision avoidance scheme can be implemented, although that would imply losing the guarantee of segregation.

V. CONCLUSIONS

We have presented an approach to the problem of segregation of multiple heterogeneous units in a robotic swarm based on the use of abstractions guided by potential functions. In contrast to previous work, we have shown a method with guaranteed convergence to segregation with multiple groups of robots. Moreover, our approach may use only local information during part of the time to segregate groups in swarms of robots.

Future work will focus on collision avoidance strategies integrated with our controller. Also, we will focus on strategies to segregate robots using only local information in any situation.

REFERENCES

[1] V. Trianni, Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots. Vol. 108. Ed. Springer Science & Business Media, 2008.

[2] M. Dorigo, et al., Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms, IEEE Robotics & Automation Magazine. Dez. 2013, pp. 60-71.

[3] C. Reynolds, Flocks, herds and schools: A distributed behavioral model, Computer Graphics, 1987, pp. 25-34.

[4] C. Belta and V. Kumar, Abstraction and Control for Groups of Robots, IEEE Transactions on Robotics, vol. 20, no. 5, Oct. 2004 pp. 865875.

[5] T. Balch and R. Alkin, Behavior-based formation control for multi-robot teams, IEEE Transactions on Robotics and Automation, v. 14, n. 6, 1998, 926-939.

[6] L. Pimenta, G. Pereira, N. Michael, R. Mesquita, M. Bosque, L. Chaimowicz and V. Kumar, Swarm Coordination based on Smoothed Particle Hydrodynamics technique, IEEE Transactions on Robotics, vol. 29, no. 2, 2013, pp. 4372-4277.

[7] V. Santos and L. Chaimowicz, Hierarchical Congestion Control for Robotic Swarms, Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp. 4372-4277.

[8] L. Pimenta, G. Pereira, M. Gonalves, N. Michael, M. Turpin, and V. Kumar, Decentralized controllers for perimeter surveillance with teams of aerial robots, Advanced Robotics, v. 27, n. 9, 2013, 697-709.

[9] G. Zhang, G. Fricke and D. Garg, Spill Detection and Perimeter Surveillance via Distributed Swarming Agents, IEEE/ASME Transactions on Mechatronics, v. 18, n. 1, Feb. 2013, pp. 121-129.

[10] P. Walker, S. Amraii, N. Chakraborty, M. Lewis and K. Sycara, Human Control of Robot Swarms with Dynamic Leaders, Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2014, pp. 1108-1113.

[11] J. Nagi, A. Giusti, L. Gambardella and G. Di Caro, Human-Swarm Interaction Using Spatial Gestures, Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2014, pp. 3834-3841.

[12] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar and M. Yim, Self-Assembly of a Swarm of Autonomous Boats into Floating Structures, Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014. p. 1234-1240.

[13] B. Remes, D. Hensen, F. van Tienen, C. Wagter, E. van der Horst, G. de Croon, Paparazzi: how to make a swarm of Parrot AR Drones fly autonomously based on GPS, International Micro Air Vehicle Conference and Flight Competition (IMAV2013), Toulouse, France, 17-20 Sept. 2013.

[14] L. Pimenta, V. Kumar, R. Mesquita, G. Pereira, Sensing and Coverage for a Network of Heterogeneous Robots. Sensing and coverage for a network of heterogeneous robots. Proceedings of the 47th IEEE Conference on Decision and Control, Dec. 2008, pp. 3947-3952

[15] R. Groß, S. Magnenat and F. Mondada, Segregation in swarms of mobile robots based on the Brazil nut effect, Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2009, pp. 4349-4355.

[16] M. Kumar, D. Garg, and V. Kumar, Segregation of heterogeneous units in a swarm of robotic agents, IEEE Transactions on Automatic Control, vol. 55, no. 3, Mar 2010, pp. 743-748.

[17] J. Chen, M. Gauci, M. J. Price and R. Groß, Segregation in swarms of e-puck robots based on the brazil nut effect, Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, 2012, pp. 163-170.

[18] V. Santos, L. Pimenta and L. Chaimowicz, Segregation of Multiple Heterogeneous Units in a Robotic Swarm, Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1112-1117.

[19] R. Olfati-Saber, Flocking for multi-agent dynamic systems: algorithms and theory, IEEE Transactions on Automatic Control, vol. 51, no. 3, 2006, pp. 401-420.

[20] C. Belta and V. Kumar, Towards Abstraction and Control for Large Groups of Robots, Control Problems in Robotics, Ed. Springer-Verlag Berlin Heidelberg, 2003, pp. 169-182.

[21] J. Slotine and W. Li, Applied nonlinear control. Vol. 60. NJ: Prentice-Hall, 1991.