

Cohesion and Segregation in Swarm Navigation

Vinicius Graciano Santos and Luiz Chaimowicz

Computer Science Department

Universidade Federal de Minas Gerais

Abstract

The use of large groups of robots in the execution of complex tasks has received much attention in recent years. Generally called robotic swarms, these systems employ a large number of simple agents to perform different types of tasks. A basic requirement for most robotic swarms is the ability for safe navigation in shared environments. Particularly, two desired behaviors are to keep robots close to their kin and to avoid merging with distinct groups. These are respectively called cohesion and segregation, which are observed in several biological systems. In this paper, we investigate two different approaches that allow swarms of robots to navigate in a cohesive fashion while being segregated from other groups of agents. Our first approach is based on artificial potential fields and hierarchical abstractions. However, this method has one drawback: it needs a central entity which is able to communicate with all robots. To cope with this problem, we introduce a distributed mechanism that combines hierarchical abstractions, flocking behaviors, and an efficient collision avoidance mechanism. We perform simulated and real experiments in order to study the feasibility and effectiveness of our methods. Results show that both approaches ensure cohesion and segregation during swarm navigation.

1 Introduction

Swarm systems generally employ a large number of simple agents to perform complex tasks. Specially in robotics, tasks such as surveillance, transportation, and exploration and mapping

of unknown environments can benefit from the use of large groups of robots. In recent years, such systems have been receiving much attention because of current advances in technology, which have been allowing the mass production of increasingly smaller robots.

Basic requirements for most robotic swarms include the ability for safe and efficient navigation. In other words, robots must be able to reach specific goals in a minimum amount of time while avoiding collisions with obstacles, teammates, and other groups of agents. One possible strategy is to keep robots close to their kin and avoid merging with other groups, since this behavior may reduce possible interferences among distinct groups during navigation [36, 42]. Two related and important properties of this strategy are *cohesion* and *segregation*. The former leads groups of robots to behave as a team, whereas the latter prevents them from mingling with different groups. These properties are naturally observed in several biological systems such as flocks of birds and schools of fishes.

In this paper, we investigate two different approaches that allow swarms of robots to navigate in a cohesive fashion while being segregated from other groups. The first one is based on artificial potential fields and hierarchical abstractions, being originally applied in [42] to avoid congestion in robot navigation. We revisit this method focusing on cohesion and segregation in a more general setting. In spite of being able to successfully keep groups segregated, the controller relies on a central unit that oversees the swarm, which may compromise the scalability of the system. To cope with this problem, we investigate a second approach based on velocity obstacles and flocking behaviors. We introduce the *Virtual Group Velocity Obstacle*: a set of forbidden velocities that can lead an agent of a particular cluster to mingle with other clusters. Furthermore, we present a series of simulated and real experiments in order to show the robustness of both approaches, and we analyze the results using a metric that measures the segregative behavior of the system along its execution [31].

This paper is organized as follows: Section 2 reviews related work in the fields of collision

avoidance, swarm control, and hierarchical abstractions. Sections 3 and 4 present our two methodologies on cohesion and segregation. Finally, Section 5 discusses our experiments and results, and Section 6 concludes the work providing suggestions for future work.

2 Related Work

Reynolds [40] was one of the first researchers who tackled the problem of realistically simulating the movement of a swarm of agents; more specifically a flock of birds known as *boids*. Basically, his approach relies on local interactions among agents within a neighboring area, which define an emergent behavior for the whole flock. Such interactions can be modeled as a special case of the social potential field method [39], an extension of the artificial potential field technique [28] that specifically deals with multi-agent systems.

Several works have focused on using artificial potential fields in conjunction with flocking rules in order to obtain specific behaviors, such as moving in formation [4], converging into shapes [9, 24], area coverage [23], shepherding [33], and so on. However, it is known that methods based on potential fields are not oscillation-free and suffer from local minima problems [30], which is an intrinsic property that can arise from the combination of potentials, especially in unknown environments.

Alternative techniques have been developed that are guaranteed to be collision-free and oscillation-free [20, 43, 46], even under nonholonomic constraints [2, 3]. They rely on the concept of Velocity Obstacles [16], an extension of the Configuration Space Obstacle [34] to a time-varying system. A Velocity Obstacle defines the set of velocities that would result in a collision between an agent and an obstacle moving at a given velocity. Thus, the robot can perform an avoidance maneuver by selecting velocities that do not belong to this set. Such approach has been widely used and extended for multi-agent navigation [1, 20, 46], even when considering uncertainties in position, shape, and velocity of the obstacles [12, 17, 43]. An

important extension was the development of the Reciprocal Velocity Obstacle by van den Berg et al. [45], who acknowledged that most works on collision avoidance had not taken into account the reciprocity that arises when obstacles are in fact other agents that can also react according to the robot’s behavior. Another extension was recently proposed for the case in which a single agent should avoid a group as a whole [21]. Despite relying on a virtual obstacle in a similar manner as our Virtual Group Velocity Obstacle, which will be explained in Section 4, the method does not focus on segregation since it is restricted to a single agent instead of a group of robots.

A different paradigm considers the whole group as a single entity in a hierarchical fashion. These entities are sometimes called virtual structures as they embody the pose and shape of a team of robots [44]. In this case, steering control laws are applied to the virtual structure in order to maneuver the robotic swarm. For instance, works presented in [15] and [44] define controllers that converge and maintain a group of robots in a rigid formation according to a known structure. Nevertheless, such methods are not easily scalable to large groups because each distance constraint among a pair of robots must be explicitly stated in order to achieve a desired formation, and these fixed geometric relations may hinder formation changes during navigation.

To address these problems, deformable structures were presented in [5] and [26] together with artificial potential fields to group and control swarms of robots. In the former, controllers were designed in order to converge the swarm into a known elliptical region, which was used to escort a vehicle convoy, whereas in the latter, *Probabilistic Roadmaps* [27] were used to plan paths for the structure in an environment with obstacles. Instead of considering a single deformable structure, some studies employed a set of structures to increase group cohesion and to simplify the path planning problem. For instance, in [32] a hierarchical sphere tree was proposed to control “crowds of robots”, and in [25] the path planned for a single agent is

extended to a corridor using the clearance along the path. Hence, it is possible to control a swarm that navigates through the corridor by changing its characteristics in a desired way.

Belta and Kumar [6] proposed a formal abstraction that allows decoupled control of the pose and shape of a team of robots. It is based on a mapping of the swarm’s configuration space to a lower dimensional manifold, whose dimension is independent of the number of robots. This work was extended in [37] to account for three dimensional swarms, and in [22] a dynamic control model was introduced for similar abstractions. Another extension was developed by Chaimowicz and Kumar [8], who studied cooperation mechanisms between multiple unmanned aerial and ground vehicles. In their work, UAVs estimated the configuration of the ground robots and sent control messages to them. Furthermore, merging and splitting behaviors were studied, since sometimes these maneuvers are necessary to overcome obstacles. Nevertheless, interactions among groups with different goals were not addressed.

Some works have specifically tackled segregation in robotic swarms: Kumar et al. [31] proposed a distributed controller that is based on the Differential Adhesion Hypothesis from cellular biology, and introduced a metric that measured segregation quantitatively. Another segregation algorithm, presented in [19], allowed mobile robots to self-organize into annular structures. A distributed controller considered robots as having different virtual sizes, and local interactions made the “larger” robots move outwards. The procedure was inspired by a granular convection phenomenon known as the Brazil Nut Effect. This work was later extended to consider real e-puck robots [10]. In spite of the interesting results, these works did not focus on the cohesion and segregation of different groups during navigation, which is the main topic of this paper. Furthermore, to the best of our knowledge, there are no other approaches that solely focus in this topic.

3 Cohesion and Segregation using Hierarchical Abstractions

We consider a set of fully actuated individual robots with dynamic model given by $\dot{\mathbf{p}}_i = \mathbf{v}_i$, $\dot{\mathbf{v}}_i = \mathbf{u}_i$, in which $\mathbf{p}_i = [x_i, y_i]^T$ is the position of robot i , \mathbf{v}_i its velocity, and \mathbf{u}_i its control input. Different groups of robots are assembled into a set Γ , in which a group $j \in \Gamma$ is modeled by a pair $(\mathcal{P}_j, \mathcal{S}_j)$ that comprises its pose and shape, respectively. For each group, this pair constitutes a control abstraction \mathcal{A}_j that we parametrize as

$$\begin{aligned} \mathcal{A}_j &= (\mathcal{P}_j, \mathcal{S}_j), \\ \mathcal{P}_j &= (\mu_j^x, \mu_j^y, \theta_j) \in SE(2), \\ \mathcal{S}_j &= (s_j^x, s_j^y) \in \mathbb{R}^2. \end{aligned} \tag{1}$$

This abstraction can be implicitly defined by the level set $C_{\mathcal{A}_j}(x, y) = 0$, which is an ellipse centered at (μ_j^x, μ_j^y) with orientation θ_j whose principal axes have length s_j^x and s_j^y . Note that these superscripts relate a specific parameter with its corresponding axis. Thus, we specify a group $j \in \Gamma$ as the set of all robots which satisfy the constraint $C_{\mathcal{A}_j}(\mathbf{p}_i) < 0$. Therefore, the curve $C_{\mathcal{A}_j}(x, y) = 0$ can be seen as a border that limits and defines a group.

In the following paragraphs, we will define two distinct control laws: the first is used by each individual robot to stay on the inside of its group and avoid collisions with nearby teammates, and the second controls the parameters of abstraction \mathcal{A}_j , such as its position while maneuvering to avoid other groups. We start by explaining the former and then we move the discussion on to the latter.

3.1 Robot's Control Law

Given a function $\phi(\mathbf{p}_i, \mathcal{A}_j)$ that maps \mathbf{p}_i to its radial distance from the border of group j , the normal function

$$f(\mathbf{p}_i, \mathcal{A}_j) = e^{-\gamma\phi^2(\mathbf{p}_i, \mathcal{A}_j)} \quad (2)$$

produces the artificial potential field, shown in Figure 1, whose maxima are located at the curve $C_{\mathcal{A}_j}(x, y) = 0$. In other words, it forms a bowl-like surface, with γ being inversely proportional to the thickness of its walls. Based on this potential, we define the following control law for each robot:

$$\mathbf{u}_i = -k_1 \sum_{j \in \Gamma} \nabla f(\mathbf{p}_i, \mathcal{A}_j) - k_2 \dot{\mathbf{p}}_i + \sum_{k \in N_i} F_r(\mathbf{p}_i, \mathbf{p}_k). \quad (3)$$

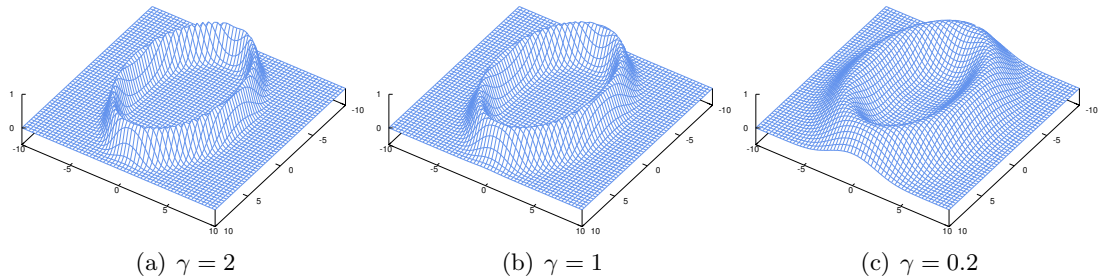


Figure 1: Artificial potential field $f(\mathbf{p}_i, \mathcal{A}_j)$ with $s_j^x = 5$ and $s_j^y = 8$ for distinct values of γ .

Constants k_1 and k_2 are positive. The first term in 3 is a summation of forces that repels robots from the border of all groups. The second term is a damping force which improves stability, and the third represents a local repulsive force that prevents collisions among robots in a given neighborhood. The set N_i consists of every robot k that is within a certain distance limit δ from robot i . Notice that the first summation can be restricted to a subset of Γ , since robot i may not be influenced by the potentials of distant groups.

3.2 Abstraction's Control Law

In order to move, rotate, and reshape groups, simple linear controllers can be applied to each component of \mathcal{A}_j . These controllers along with the ones defined by (3) establish a hierarchy in

which robots are implicitly controlled according to the high level abstractions. In the following discussion, we present a hierarchical controller that maintains distinct groups segregated during navigation.

In the beginning, it is important to note that controller (3) forces agents to avoid any intersection areas among groups, given that robots do not initially lie in these areas. For example, this behavior can be better understood when a collision between two groups takes place: the robots within a group will be repelled by the border of the other group. Thus, we exploit this feature in order to keep robots segregated.

The general idea of the segregation algorithm is to take advantage of the geometric features of the virtual structure $C_{\mathcal{A}_j}(x, y) = 0$ to create repulsive forces among groups in order to divert them from possible areas where merging may happen.

Given two groups $m, n \in \Gamma$, let \mathbf{c} be the centroid of the intersection points between the curves $C_{\mathcal{A}_m}(x, y) = 0$ and $C_{\mathcal{A}_n}(x, y) = 0$. The repulsive force F_{rep} among groups should be directly proportional to the penetration depth of \mathbf{c} in relation to each group. For \mathcal{A}_m , this depth is simply given by the radial distance $\phi(\mathbf{c}, \mathcal{A}_m)$, and the force acting on this group can be written as

$$F_{\text{rep}}(\mathcal{A}_m, \mathcal{A}_n) = \begin{cases} 0, & \text{if there are no intersection points;} \\ \phi(\mathbf{c}, \mathcal{A}_m) \frac{(\boldsymbol{\mu}_m - \mathbf{c})}{\|\boldsymbol{\mu}_m - \mathbf{c}\|}, & \text{otherwise;} \end{cases} \quad (4)$$

in which $\boldsymbol{\mu}_m = [\mu_m^x, \mu_m^y]^T$.

Equation (4) tries to minimize the intersection area between the two groups. However, if they are moving in opposite directions, it is possible that these forces may cancel out the force that drives abstraction \mathcal{A}_j towards its goal. Therefore, we employ an artificial vortex field [13] in order to steer them away during a collision, i.e., a rotational force F_{rot} is defined such that it is perpendicular to F_{rep} . Since (4) defines a vector field $F_{\text{rep}} = (F_x, F_y)$, we simply

set $F_{\text{rot}} = (-F_y, F_x)$ to satisfy the orthogonality constraint. Thus, we express the force that deviates group \mathcal{A}_m from \mathcal{A}_n as

$$D(\mathcal{A}_m, \mathcal{A}_n) = F_{\text{rep}}(\mathcal{A}_m, \mathcal{A}_n) + F_{\text{rot}}(\mathcal{A}_m, \mathcal{A}_n). \quad (5)$$

We also consider a simple dynamic model for each abstraction, in which we control only its position $\boldsymbol{\mu}$. In other words, orientations and scalings remain constant throughout all time steps.

$$\ddot{\boldsymbol{\mu}}_m = k_3 \sum_{n \in \Gamma} D(\mathcal{A}_m, \mathcal{A}_n) - k_4 \dot{\boldsymbol{\mu}}_m + F_{\text{goal}}(\boldsymbol{\mu}_m) \quad (6)$$

Equation 6 requires an attractive force $F_{\text{goal}}(\boldsymbol{\mu}_m)$ that drives the group toward its goal position, such as the usual quadratic potential [11]. In Figure 2, we show the overall behavior of controllers (3) and (6) when coupled together. Note that the norm of F_{rot} must be set to zero when group m is near its goal, otherwise an endless loop of circular motion can occur if another group tries to stop at the same place simultaneously.

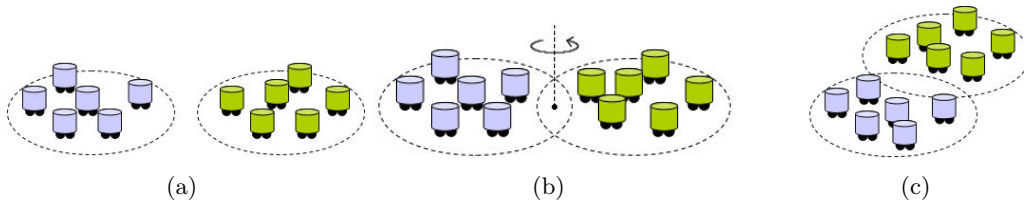


Figure 2: Execution steps of the segregation algorithm using hierarchical abstractions.

This approach requires a central unit which controls and broadcasts the virtual abstractions' parameters to all robots. This can be obtained, for instance, through the use of a small group of aerial robots that control these abstractions and communicate with ground agents, as it was proposed in [8]. Although this type of architecture was shown to be feasible [7], a distributed approach would be a better suited solution in order to avoid errors in the central unit that could lead to a complete system failure.

4 Cohesion and Segregation using Velocity Obstacles

As an alternative to the centralized hierarchical abstraction method, we propose a distributed approach that relies on velocity obstacles. We start by briefly reviewing its core concepts and then we introduce our methodology.

4.1 Velocity Obstacles

Let A and B be two robots moving on the plane. The velocity obstacle $VO_B^A(\mathbf{v}_B)$ of B to A is defined in the velocity space of A as the set of all velocities that will result in a collision between robots A and B at some instant in time [16]. To formally define it, we specify $\lambda(\mathbf{p}, \mathbf{v})$ as a ray starting at \mathbf{p} heading in the direction of \mathbf{v} and $B \oplus -A$ as the Minkowski sum of B and $-A$, in which $-A$ represents robot A reflected about its reference point.

$$\lambda(\mathbf{p}, \mathbf{v}) = \{\mathbf{p} + t\mathbf{v} \mid t \geq 0\}. \quad (7)$$

With these definitions, we can say that a velocity $\mathbf{v}_A \in VO_B^A(\mathbf{v}_B)$ if and only if the ray starting at \mathbf{p}_A heading in the direction $\mathbf{v}_A - \mathbf{v}_B$ intersects $B \oplus -A$. Therefore, the full set of velocities that specifies a Velocity Obstacle can be denoted as

$$VO_B^A(\mathbf{v}_B) = \{\mathbf{v} \mid \lambda(\mathbf{p}_A, \mathbf{v} - \mathbf{v}_B) \cap (B \oplus -A) \neq \emptyset\}. \quad (8)$$

This set has an interesting property: if A selects a velocity outside the $VO_B^A(\mathbf{v}_B)$ and B maintains its current velocity, it is guaranteed that a collision will not occur between them [16]. We show in Figure 3(a) a diagram of the Velocity Obstacle in a system with two circular mobile robots. As it can be seen, $VO_B^A(\mathbf{v}_B)$ is a cone with its apex at (\mathbf{v}_B) .

The use of velocity obstacles can lead to oscillation issues when dealing with multi-robot systems. To address this problem, the Reciprocal Velocity Obstacle was developed [45]. Es-

essentially, the RVO comprises all velocities that are the average of the robot's current velocity and a velocity within its Velocity Obstacle. Formally, we have

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v} \mid 2\mathbf{v} - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\}, \quad (9)$$

which can be seen as the cone $VO_B^A(\mathbf{v}_B)$ translated such that its apex lies at the mean of \mathbf{v}_A and \mathbf{v}_B , as shown in Figure 3(b). Assuming that B behaves reciprocally, if A selects a velocity outside the set $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$, that is the closest to its prior velocity, then navigation is guaranteed to be collision-free and oscillation-free [45].

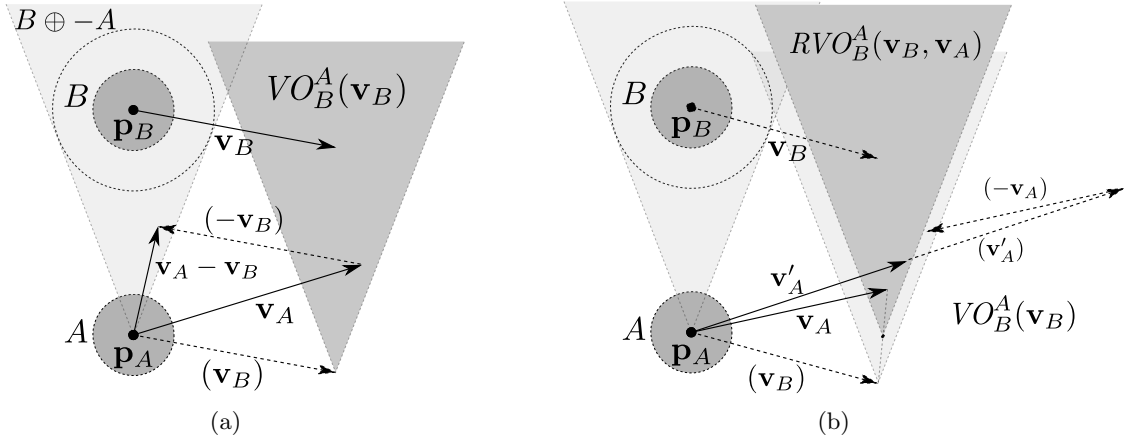


Figure 3: (a) Velocity Obstacle $VO_B^A(\mathbf{v}_B)$. (b) Reciprocal Velocity Obstacle $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$. (Adapted from [45]).

4.2 Virtual Group Velocity Obstacle

Our main objective is to safely navigate large groups of robots in a shared environment while maintaining cohesion and segregation among groups. In this section, we extend the Velocity Obstacle framework with flocking behaviors and hierarchical abstractions to achieve our goal.

At first, we need to redefine what we mean by a group. In Section 3, we specified it as the set of robots that lie inside the closed curve $C_{\mathcal{A}_j}(x, y) = 0$. Generally, this means that an agent may belong to more than one group if it is located in areas where these curves overlap. In this section, we consider that robots are assembled together into a set of disjoint groups

$\Gamma = \Gamma_1 \cup \Gamma_2 \cup \dots \cup \Gamma_N$, in which $\forall j, k : j \neq k \rightarrow \Gamma_j \cap \Gamma_k = \emptyset$. Furthermore, we assume that a robot can infer the respective group of any other agent. This can be done by using onboard sensors such as cameras or by broadcasting identifiers. Alternatively, groups can be formed by any kind of clustering algorithm.

Let $\Phi_k \subseteq \tau_k$ be the set of robots belonging to group τ_k that are within the neighborhood N_i such that $i \notin \tau_k$, i.e., all agents of a particular group that are inside the sensing radius of robot i . Furthermore, we declare $\mathbf{p}(\Phi_k)$ and $\mathbf{v}(\Phi_k)$ as the average position and the average velocity of group Φ_k , respectively. In order to achieve segregation, we introduce a virtual Velocity Obstacle that is responsible for blocking velocities which may lead groups to merge. Specifically, we denote this virtual obstacle as the *Virtual Group Velocity Obstacle* (VGVO), which is shown in Figure 4.

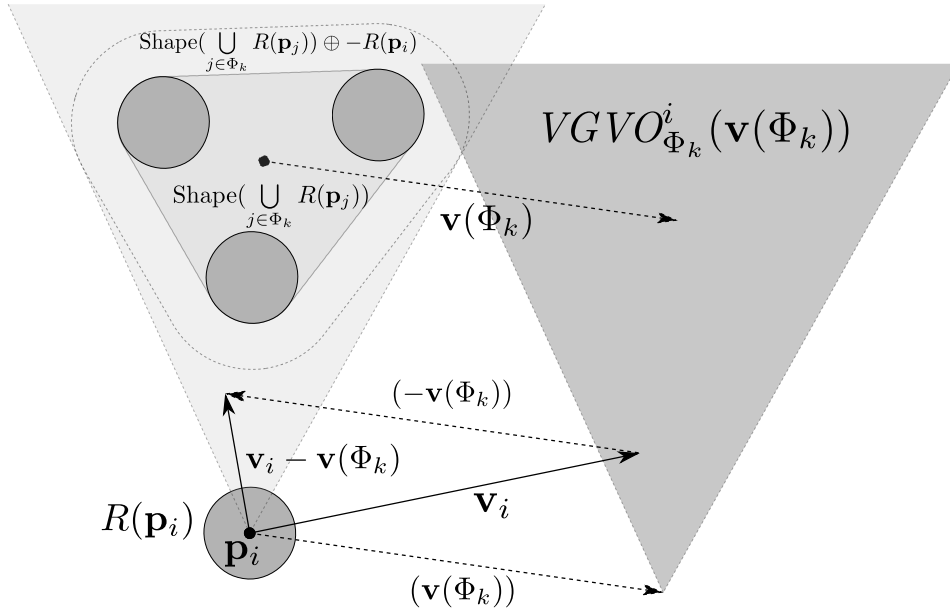


Figure 4: The Virtual Group Velocity Obstacle $\text{VGVO}_{\Phi_k}^i(\mathbf{v}(\Phi_k))$.

The VGVO is a simple concept: robot i senses the presence of every robot j within the neighborhood N_i and builds the shape of each group of robots, with the exception of its own. These shapes are considered as virtual obstacles in the workspace of robot i that move with the respective average velocity of the group that has been used during the building process. Thus,

a virtual Velocity Obstacle can be built for each shape in order to define the set of velocities that will lead the robot to merge with a different group, assuming that the latter maintains its current average velocity.

The Virtual Group Velocity Obstacle of robot i induced by group Φ_k can be written as

$$VGVO_{\Phi_k}^i(\mathbf{v}(\Phi_k)) = \{\mathbf{v} \mid \lambda(\mathbf{p}_i, \mathbf{v} - \mathbf{v}(\Phi_k)) \cap C(\mathbf{p}_i, \Phi_k) \neq \emptyset\}, \quad (10)$$

$$C(\mathbf{p}_i, \Phi_k) = \text{Shape}\left(\bigcup_{j \in \Phi_k} R(\mathbf{p}_j)\right) \oplus -R(\mathbf{p}_i), \quad (11)$$

in which $\text{Shape}(Q)$ is the shape of the set of points Q , and $R(\mathbf{p}_i)$ denote the set of points which represent robot i in its workspace. The former can be represented as the smallest enclosing disc, the convex hull, or the more general class of α -shapes [14].

Equation (11) refers to the idea of the hierarchical abstraction paradigm (Sections 2 and 3), in which the whole group is considered as a single entity. In this case, we abstract a whole group as a single entity that moves according to the average velocities of its underlying robots. Thus, single robots navigate using the RVO in conjunction with the VGVO. The former guarantees a collision-free navigation, and the latter maintains the segregative behavior. However, these two mechanisms cannot ensure cohesion, i.e., the ability of agents to stay together as a team. We will account for this using flocking rules during the velocity selection phase, as we will discuss in the next section.

4.3 Velocity Selection

An optimization problem must be solved to select inputs when dealing with Velocity Obstacles, and several distinct approaches have been developed [16, 20, 43, 45, 46]. In this work, we achieve cohesion by extending the velocity selection process presented in [45] to account for flocking rules. Basically, the method fast samples the set of admissible velocities and selects the best one according to an utility function. Although other methods have been developed to improve

cohesion [29], flocking rules are widely employed in swarm systems and it is interesting to couple them with the velocity obstacle framework.

Let $\mathbf{v}_i^{\text{pref}}$ be the preferred velocity of robot i , such as the vector pointing in the direction of its goal with magnitude equal to the maximum allowed speed. In each iteration, velocities are sampled using an uniform distribution from the set of admissible velocities

$$AV^i(\mathbf{v}_i) = \{(\mathbf{v} \mid \|\mathbf{v}\| < v_i^{\text{max}}) \wedge (\|\mathbf{v} - \mathbf{v}_i\| < a_i^{\text{max}} \Delta t)\}, \quad (12)$$

in which v_i^{max} and a_i^{max} are the maximum speed and maximum acceleration of robot i , respectively, and Δt is the time step of the system. This set comprises all reachable velocities from \mathbf{p}_i given the robot's kinematic and dynamic constraints.

Among the sampled set of admissible velocities, robot i should be able to selected a velocity $\mathbf{v}_i^{\text{new}}$ that lies outside the union of all VGVOs and RVOs, as shown in Figure 5. However, as the environment may become crowded to the point that no admissible velocities exist, the robot is allowed to selected a velocity that belongs to a velocity obstacle, but the choice is penalized according to the following function:

$$\mathbf{v}_i^{\text{flock}} = \mathbf{v}_i^{\text{pref}} + \alpha(\mathbf{v}(\Phi_k) - \mathbf{v}_i) + \beta(\mathbf{p}(\Phi_k) - \mathbf{p}_i) \quad (13)$$

$$P_i(\mathbf{v}) = \frac{w}{T_i(\mathbf{v})} + \left\| \mathbf{v}_i^{\text{flock}} - \mathbf{v} \right\|, \quad (14)$$

with $i \in \Gamma_k$. In the above equation, α weighs the alignment of the new velocity to the average velocity of teammates, β weighs convergence of robot i to the centroid of its group, and w regulates the avoidance behavior between sluggishness and aggressiveness. Function $T_i(\mathbf{v})$ is the expected time to collision, which is computed by minimizing the solutions of the set of ray intersection equations induced by (8) and (9). Thus, robot i selects the velocity $\mathbf{v}_i^{\text{new}}$ that

minimizes the penalty function P_i over the sampled set $S \subseteq AV^i(\mathbf{v}_i)$.

$$\mathbf{v}_i^{\text{new}} = \underset{\mathbf{v} \in S}{\operatorname{argmin}} P_i(\mathbf{v}) \quad (15)$$

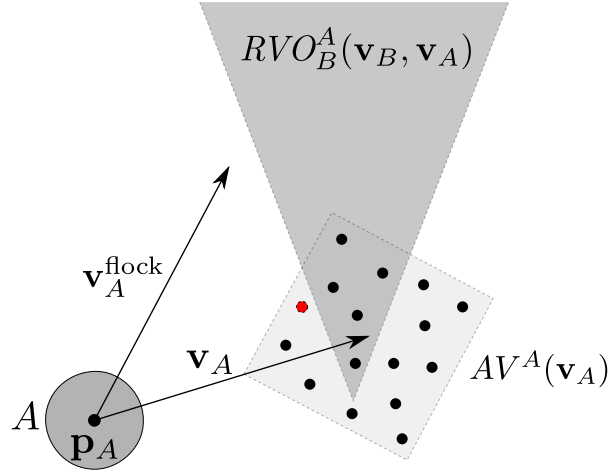


Figure 5: Sampling-based velocity selection. Admissible velocities which were sampled are represented by small circles. The red sample is chosen as it minimizes the penalty function.

5 Experiments

In this section, we compare the hierarchical abstraction (Section 3) to the virtual group velocity obstacle (Section 4) in terms of their segregative behavior as well as the time taken by each group to reach their destination. We evaluate both of these using a metric that compares the average distances among robots in different groups of the swarm [31].

Additionally, we present experiments with two other methodologies for swarm navigation: basic attractive/repulsive potential fields [28] and reciprocal velocity obstacles [45]. The basic artificial potential field approach consists in each robot being attracted towards its goal while being repelled by nearby robots. In our implementation, we have employed potential functions such as the ones presented in [11]. Furthermore, we implemented the RVO algorithm according to its description in Section 4.1, in which the mechanism of Section 4.3 was used to select

velocities at each iteration, and flocking behaviors were inhibited by setting constants α and β to zero. We use these methods in order to show how the chosen metric reflects the behavior of controllers that do not consider segregation. More detailed comparisons of these two with our proposed controllers can be found in [41] and [42].

5.1 Simulations

Each simulation consists of a scenario where robots are evenly partitioned into distinct groups. Initially, agents are randomly positioned according to a normal distribution into a circular area around the initial position of their group. Afterwards, groups are commanded to swap their positions. All robots have a limited sensing range as well as restrictions concerning their maximum speeds and accelerations. Although our hierarchical controller requires a centralized unit that broadcasts the abstraction’s parameters, robots avoid collisions among themselves by solely relying on local sensing. Moreover, in order to properly reflect the mathematical definition of the VGVO, we have used α -shapes [14] as a shape descriptor of each group. However, the aperture of the VGVO can be completely described by the positions of the two agents that maximize their radial distance from each other in the frame of reference of robot i . This can be easily seen in Figure 4, in which the VGVO would have the same aperture if the middle agent were removed from the depicted group. Therefore, this characteristic can be used to optimize the implementation.

Figure 6 shows two groups of one hundred robots swapping their positions using all four presented methods. As can be seen, the hierarchical abstraction (6(c)) and the VGVO (6(d)) are capable of maintaining cohesion and segregation. On the other hand, neither potential fields (6(a)) nor RVOs (6(b)) achieve segregation since these methods were not developed with this intent. In this specific scenario, we can observe that navigation based on the RVO tends to form lines of robots, whereas the VGVO, in conjunction with flocking behaviors, stretches groups into elongated formations. Similarly, the use of repulsive/attractive potential fields

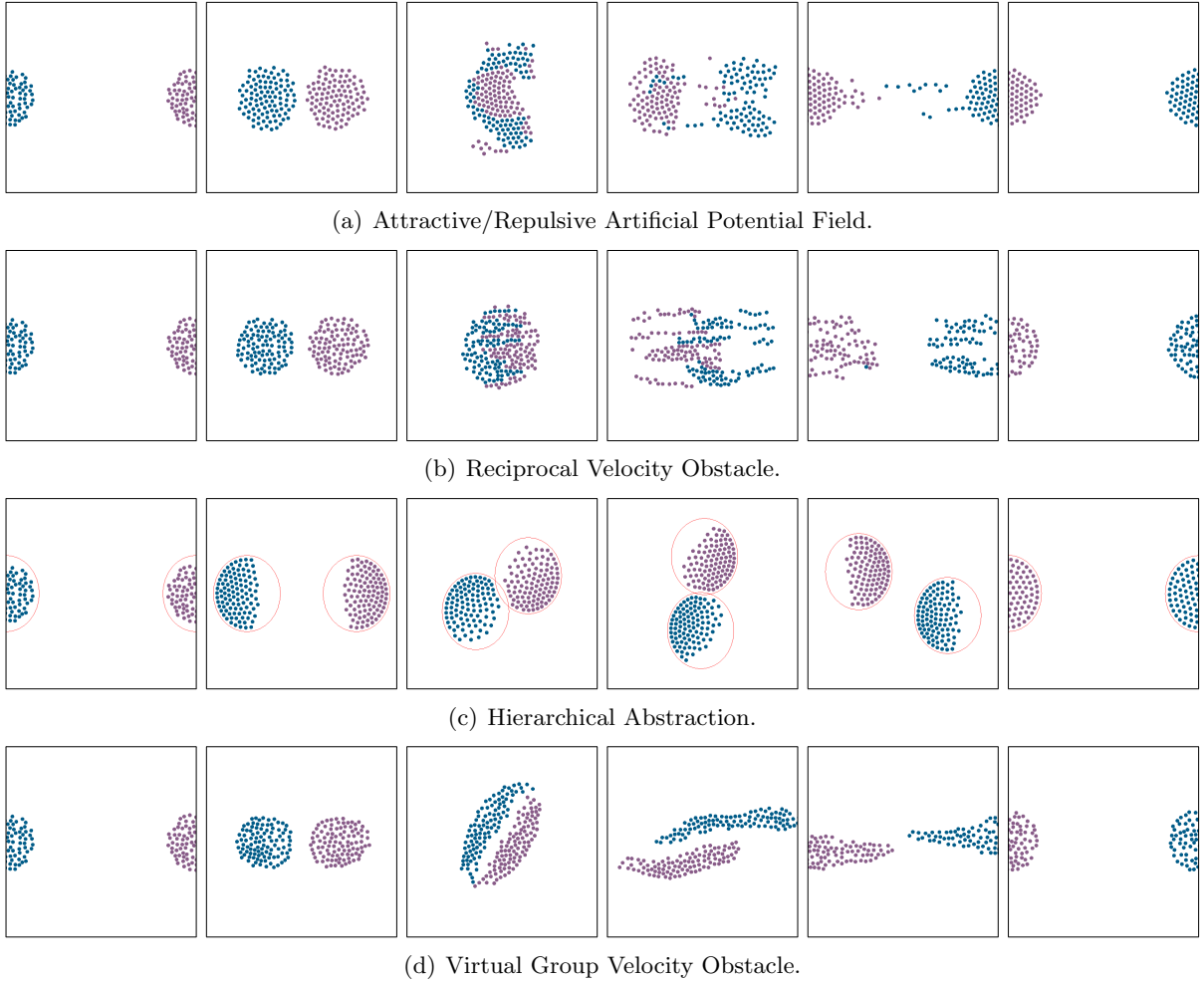


Figure 6: Behavioral comparison among controllers with two hundred robots evenly distributed into two groups using local sensing.

leads groups to directly crash into each other, while the hierarchical abstraction, in spite of moving a group toward another, prevents agents from mingling with distinct groups as a result of the avoidance behavior of its virtual structure.

As mentioned, a formal way of measuring segregation among groups of agents has recently been proposed [31]. Two groups Γ_A and Γ_B are said to be segregated if the average distance among robots in the same group is less than the average distance among robots in distinct groups. In other words, the following restriction must hold

$$(d_{\text{avg}}^{AA} < d_{\text{avg}}^{AB}) \wedge (d_{\text{avg}}^{BB} < d_{\text{avg}}^{AB}), \quad (16)$$

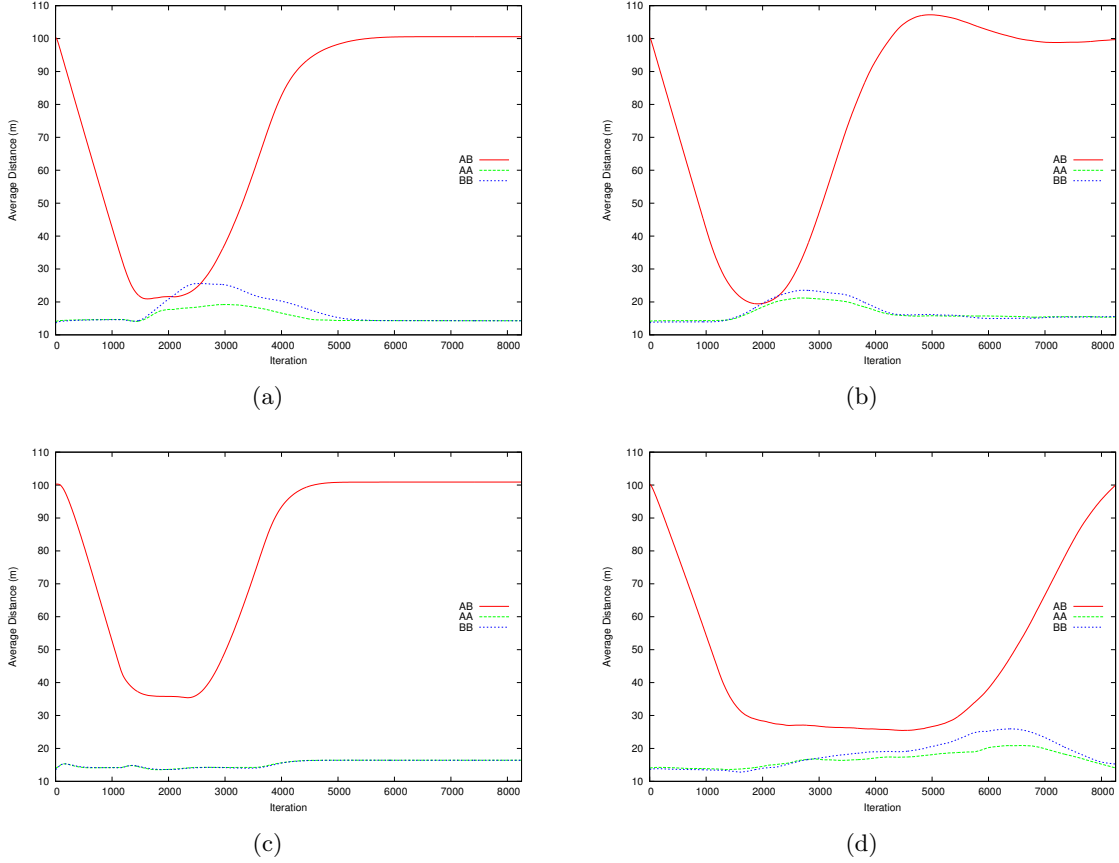


Figure 7: Segregative behavior analysis for two hundred robots evenly distributed into two groups that swap positions. (a) Attractive/Repulsive Artificial Potential Field. (b) Reciprocal Velocity Obstacle. (c) Hierarchical Abstraction. (d) Virtual Group Velocity Obstacle.

in which d_{avg}^{AB} is the average distance among robots in group Γ_A and Γ_B .

In Figure 7, we depict these average distances with regards to the presented simulations. As can be seen in 7(c) and 7(d), our controllers have successfully achieved the segregative property in the sense of constraint (16). On the other hand, in Figures 7(a) and 7(b) segregation is not achieved: the constraint is violated since there are intersection points between the curves d_{avg}^{AB} and d_{avg}^{BB} .

Another important piece of information that can be extracted from Figure 7 is the total amount of time required to complete the task. When curve d_{avg}^{AB} returns to its initial value, it means that both groups have swapped their average positions. In other words, the total navigation time is related to the concavity of this curve. Hence, we can consider that robots complete their task when the latter is met, instead of requiring that all goals are reached. This

condition is interesting because, in its terms, the convergence to the goal does not interfere with cohesion and segregation during navigation, which are the focus of our analysis. With this in mind, we can see that the RVO is the fastest approach, followed by the hierarchical abstraction, artificial potential fields, and the VGVO. The performance loss of the VGVO happens because the second and third terms of (13) play a damping role. Additionally, robots have a tendency to select safer velocities when maintaining the flock, i.e., they prioritize slower speeds during the velocity selection specified by (15).

We complement these results by presenting another set of simulations in a similar scenario, where we partitioned two hundred robots into four groups. Figure 8 illustrates these experiments. In 8(a) a large cluster is formed in the center of the environment, which slowly dissipates as robots reach their target. We observe the same behavior in the use of RVO, but its cluster tends to move around. A symmetrical avoidance behavior was achieved in the experiment of Figure 8(c) because we have used circles as the shapes of the virtual structures. Both simulations 8(c) and 8(d) have achieved cohesion and segregation in the sense of (16). We do not show the average distance plots for these experiments since the combination of all groups would result in an excessive 10 curves per figure. Nevertheless, results were similar to the ones obtained in Figure 7, i.e., both the hierarchical abstraction and the VGVO achieved segregation, and robots took a longer time to finish the task with the latter than with the former.

5.2 Real Robots

We have also validated our results in proof-of-concept experiments with real robots. Such experiments are important in order to show the feasibility of the algorithms in real scenarios, where all uncertainties caused by sensing and actuation errors may have an important role on results. We used a set of twelve e-puck robots [38], which are small-sized differential robots equipped with a ring of 8 IR sensors for proximity sensing and a set of LEDs for displaying

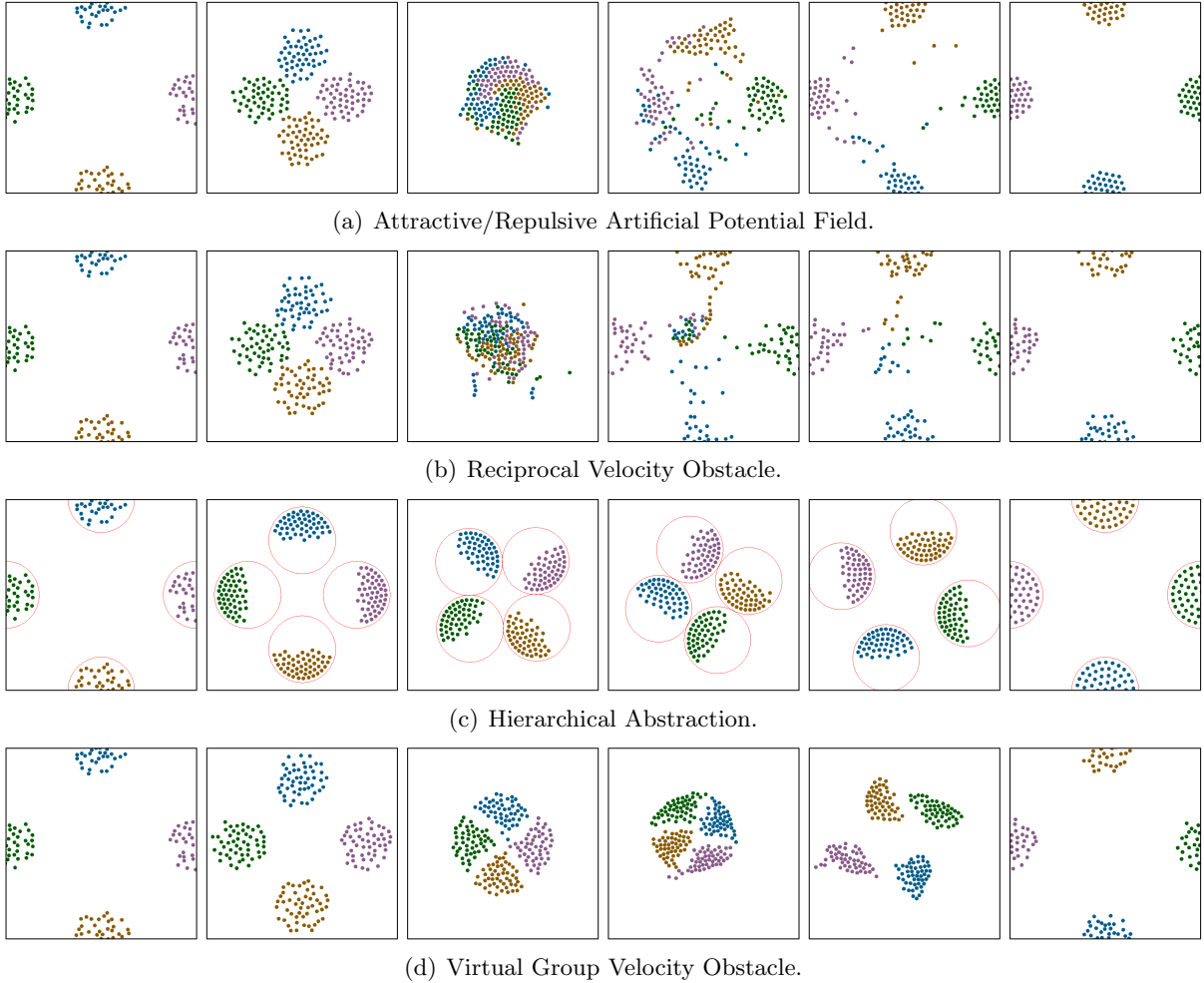
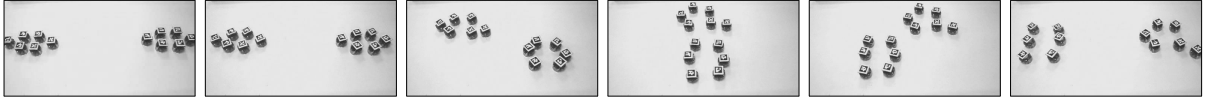


Figure 8: Behavioral comparison among controllers with two hundred robots evenly distributed into four groups using local sensing.

status. A bluetooth wireless interface allows local communication among robots and also with a remote computer. We controlled these robots through Player [18], a well known framework for robot simulation and programming.

In order to estimate the configuration of all robots, we used a swarm localization architecture based on fiduciary markers and overhead cameras. Computers process the captured images from these cameras and determine the position of all robots in a common frame of reference. Afterwards, control inputs are calculated according to our approaches and broadcast to the swarm. Additionally, we implemented a virtual sensor to detect neighboring agents because the e-puck’s IR sensors have a very small range. Furthermore, to account for nonholonomic constraints, we transformed input velocities according to the approach presented in [35].

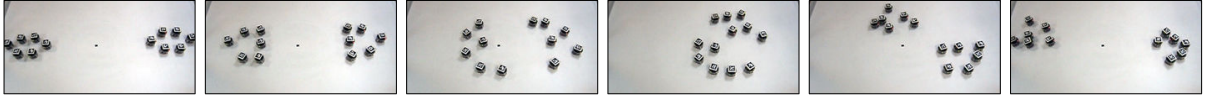


(a) Two robotic groups.

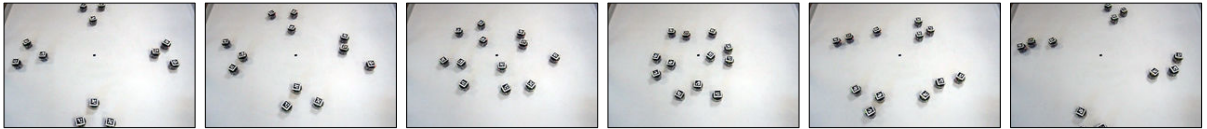


(b) Four robotic groups.

Figure 9: Real execution of the hierarchical abstraction algorithm with different group sizes.



(a) Two robotic groups.



(b) Four robotic groups.

Figure 10: Real execution of the VGVO algorithm with different group sizes.

Figures 9 and 10 show snapshots from executions of the hierarchical abstraction and VGVO approaches, respectively. We can visually inspect that the experimental results are similar to the simulations, i.e., robots maintain cohesion and segregation during navigation. We observed that average distances follow the trend shown in Figure 7: the average distance between robots of same groups is always less than the average distance among robots of distinct groups. Although these experiments indicate that our controllers may work reasonably well to ensure cohesion and segregation, we emphasize that they are proof-of-concepts only, and more experiments are needed in order to fully evaluate the proposed approach in real swarm systems.

5.3 Discussion

Both of our algorithms require careful tuning of constants. Regarding the hierarchical abstraction, values must be adjusted so that the virtual ellipses respect the speed and velocity constraints of the ground robots. Also, constants k_1 and γ must be adjusted according to the

repulsion forces among agents, in a way that their summation will not cause robot i to leave its group. Additionally, the ϕ function explicitly defines the shape of the group, so it can be changed to account for other structures, such as a triangle or a rectangle. Finally, constants k_3 and k_4 must be tuned in order to ensure that the abstraction does not surpass robots during its movement.

The Velocity Obstacle framework is known for allowing high-speed navigation in multi-robot scenarios, but, when trying to ensure cohesion using flocking rules, we have seen that robots tend to select slower speeds with our method, as evidenced in Figure 7(d). Thus, this result directly impacts the tuning process of all constants in our approach. For example, given a high value for α in (13), robots will quickly align their velocities to the average velocity of their neighbors, which in turn can lead to overshoot goal positions as well as increase the chances of collisions with single agents moving in high-speeds. Similarly, a high value for β can lead robots into tightly aggregated groups, which makes agents prefer slower speeds because most higher speeds will be inside the velocity obstacles. Moreover, by setting $\alpha = \beta = 0$, the velocity selection scheme is reduced to the original RVO method [45], as the flocking behavior is discarded.

We can see evidence of these discussed problems in Figure 7(b), in which there is a noticeable overshoot of the average distance among robots in different groups. The same has happened in the experiment of Figure 7(d), but it is not shown since this could visually skew the total navigation time comparison, which is one of our main concerns besides segregation. For both algorithms, this issue arises because there is no damping over the robot's input. Moreover, the velocity matching term of (13) actually worsens the problem, and it is easy to see that agents may leave the goal when trying to match their velocities. Therefore, parameters α and β must be chosen with care since higher values can compromise the swarm behavior over the individual behavior, i.e., matching velocities over convergence to the goal. An interesting approach would

be to completely dismiss the group behavior as soon as a robot is close to its goal.

Finally, for reference value, we present in Table 1 the values of all constants used in the simulated experiments shown in Figure 6.

α	β	w	v_i^{\max}	a_i^{\max}	Δt
1	0.2	35	6	120	0.01

k_1	k_2	k_3	k_4	γ	δ	v_i^{\max}	a_i^{\max}	Δt
5000	0.8	0.01	0.8	0.8	20	6	120	0.01

Table 1: Constants used in the experiments shown in Figure 6.

6 Conclusion

In this work, we have proposed two different methods that allow swarms of robots to navigate in a cohesive fashion while maintaining segregation. We based our first method on a high-level abstraction that groups robots using artificial potential fields. In this manner, individual robots are implicitly controlled by changing the parameters of this abstraction. By considering its geometrical features, which define a virtual structure, we maintained robot segregation by relying on virtual forces generated from the intersection points between a pair of structures. In our second technique, we have introduced a novel concept: the Virtual Group Velocity Obstacle, a virtual obstacle that prevents single agents from mingling into other robotic groups, which in turn ensures segregation. Particularly, the VGVO resembles ideas from the hierarchical abstraction paradigm, in which groups are abstracted into single entities. We have maintained group cohesion by coupling the velocity obstacle framework with flocking behaviors. More specifically, we biased the robot’s preferred velocity in order to account for flocking rules by choosing a proper utility function during the velocity selection phase.

We performed several experiments in simulated and real scenarios, and the results demonstrate the effectiveness of the proposed approaches. In spite of this evidence, there are still opportunities for improvements. For instance, abstractions could employ complex avoidance

behaviors, such as expansions, contractions, and rotations, as these maneuvers may decrease the total navigation time and possibly allow us to deal with static obstacles. Furthermore, one of the downsides of our second approach is its performance in relation to time. This may be improved by properly balancing the shared effort among groups, in the same manner as the ORCA [46] algorithm does for a single pair of robots, or by relying on different methods for preferred velocity biasing [21] or achieving cohesion [29]. Further investigation along these lines may lead to interesting results that could further extend the velocity obstacle framework.

Acknowledgments

This work was partially supported by CAPES, CNPq, and FAPEMIG. We would like to thank Dorca Lee for reviewing this manuscript.

References

- [1] Y. Abe and M. Yoshiki. Collision avoidance method for multiple autonomous mobile agents by implicit cooperation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1207 – 1212, 2001.
- [2] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart. Reciprocal collision avoidance for multiple car-like robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 360 –366, may 2012.
- [3] J Alonso-Mora, A Breitenmoser, M Ruffi, P Beardsley, and R Siegwart. Optimal reciprocal collision avoidance for multiple non-holonomic robots. In *Proceedings of the 10th International Symposium on Distributed Autonomous Robotic Systems*, November 2010.

- [4] T. Balch and M. Hybinette. Social potentials for scalable multirobot formations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 73–80, 2000.
- [5] L.E. Barnes, M.A. Fields, and K.P. Valavanis. Swarm formation control utilizing elliptical surfaces and limiting functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1434–1445, December 2009.
- [6] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, October 2004.
- [7] C. Belta, G.A.S. Pereira, and V. Kumar. Control of a team of car-like robots using abstractions. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1520 – 1525, dec. 2003.
- [8] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. In *7th International Symposium on Distributed Autonomous Robotic Systems*, June 2004.
- [9] L. Chaimowicz, N. Michael, and V. Kumar. Controlling swarms of robots using interpolated implicit functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2498–2503, 2005.
- [10] Jianing Chen, Melvin Gauci, Michael J. Price, and Roderich Groß. Segregation in swarms of e-puck robots based on the brazil nut effect. In *AAMAS*, pages 163–170, 2012.
- [11] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.

- [12] Daniel Claes, Daniel Hennes, Karl Tuyls, and Wim Meeussen. Multi-robot collision avoidance with localization uncertainty. In *Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012.
- [13] C. de Medio and G. Oriolo. Robot obstacle avoidance using vortex fields. In *Advances in Robot Kynematics*, pages 227–235. S. Stifter and J. Lenarčič (Eds.), Springer-Verlag, 1991.
- [14] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, July 1983.
- [15] M. Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, December 2001.
- [16] Paolo Fiorini and Zvi Shillert. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17:760–772, 1998.
- [17] C. Fulgenzi, A. Spalanzani, and C. Laugier. Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1610 –1616, April 2007.
- [18] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.
- [19] R. Groß, S. Magnenat, and F. Mondada. Segregation in swarms of mobile robots based on the brazil nut effect. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4349–4356, October 2009.
- [20] Stephen. J. Guy, Jatin Chhugani, Changkyu Kim, Nadathur Satish, Ming Lin, Dinesh Manocha, and Pradeep Dubey. Clearpath: highly parallel collision avoidance for multi-

- agent simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187, New York, NY, USA, 2009. ACM.
- [21] Liang He and Jur van den Berg. Meso-scale planning for multi-agent navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [22] S.P. Hou, C.C. Cheah, and J.J.E. Slotine. Dynamic region following formation control for a swarm of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1929–1934, May 2009.
- [23] A Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems*, 2002.
- [24] M. A. Hsieh, V. Kumar, and L. Chaimowicz. Decentralized controllers for shape generation with robotic swarms. *Robotica*, 26(5):691–701, 2008.
- [25] A. Kamphuis and M. H. Overmars. Finding paths for coherent groups using clearance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 19–28, 2004.
- [26] A. Kamphuis and M.H. Overmars. Motion planning for coherent groups of entities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3815–3822, May 2004.
- [27] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 12(4):566–580, August 1996.

- [28] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 500 – 505, March 1985.
- [29] Andrew Kimmel, Andrew Dobson, and Kostas Bekris. Maintaining team coherence under the velocity obstacle framework. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 247–256, 2012.
- [30] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1404, 1991.
- [31] M. Kumar, D.P. Garg, and V. Kumar. Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3):743–748, March 2010.
- [32] Tsai-Yen Li and Hsu-Chi Chou. Motion planning for a crowd of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 4215–4221, September 2003.
- [33] J.-M. Lien, O.B. Bayazit, R.T. Sowell, S. Rodriguez, and N.M. Amato. Shepherding behaviors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4159–4164, 2004.
- [34] T. Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, C-32(2):108 –120, feb. 1983.
- [35] A. De Luca, G. Oriolo, and M. Vendittelli. Stabilization of the unicycle via dynamic feedback linearization. In *6th IFAC Symposium on Robot Control*, pages 397 – 402, 2000.

- [36] L.S. Marcolino and L. Chaimowicz. Traffic control for a swarm of robots: Avoiding group conflicts. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1949–1954, October 2009.
- [37] N. Michael, C. Belta, and V. Kumar. Controlling three dimensional swarms of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 964–969, May 2006.
- [38] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klapotocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65, 2009.
- [39] John H. Reif and Hongyan Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27(3):171–194, 1999.
- [40] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, pages 25–34, 1987.
- [41] Vinicius Santos, Mario Campos, and Luiz Chaimowicz. On segregative behaviors using flocking and velocity obstacles. In *Proceedings of the 11th International Symposium on Distributed Autonomous Robotic Systems*, November 2012.
- [42] Vinicius Santos and Luiz Chaimowicz. Hierarchical congestion control for robotic swarms. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4372–4377, September 2011.
- [43] J. Snape, J. van den Berg, S.J. Guy, and D. Manocha. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27(4):696–706, August 2011.

- [44] Kar-Han Tan and M.A. Lewis. Virtual structures for high-precision cooperative mobile robotic control. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 132–139, November 1996.
- [45] J. van den Berg, Ming Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1928–1935, May 2008.
- [46] Jur van den Berg, Stephen J. Guy, Ming C. Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *The 14th International Symposium on Robotics Research*. Springer Tracts in Advanced Robotics, May 2009.