COMPORTAMENTOS SEGREGATIVOS EM ENXAMES ROBÓTICOS

VINICIUS GRACIANO SANTOS

COMPORTAMENTOS SEGREGATIVOS EM ENXAMES ROBÓTICOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Luiz Chaimowicz

Belo Horizonte

Janeiro de 2014

VINICIUS GRACIANO SANTOS

SEGREGATIVE BEHAVIORS IN SWARM SYSTEMS

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Luiz Chaimowicz

Belo Horizonte

January 2014

Santos, Vinicius Graciano

S237s Segregative Behaviors in Swarm Systems / Vinicius Graciano Santos. — Belo Horizonte, 2014
xvi, 60 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais. Departamento de Ciência da Computação.
Orientador: Luiz Chaimowicz

1. Computação — Teses. 2. Robótica — Teses. I. Orientador. II. Título.

CDU 519.6*82.9 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Comportamentos segregativos em enxames robóticos

VINICIUS GRACIANO SANTOS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Luiz Chaimowicz - Orientador
Departamento de Ciência da Computação - UFMG

Profa. Anna Helena Reali Costa
Departamento de Engenharia de Computação e Sistemas Digitais - USP

Prof. Guilherme Augusto Silva Pereira
Departamento de Engenharia Elétrica - UFMG

Prof. Luciano Cunha de Araújo Pimenta
Departamento de Engenharia Eletrônica - UFMG

Belo Horizonte, 31 de janeiro de 2014.

Resumo

Robótica de enxames é o estudo de sistemas multiagentes cujos robôs são relativamente simples e possuem capacidades limitadas. Esses sistemas usualmente dependem de propriedades como robustez, flexibilidade e escalabilidade para cumprir tarefas complexas em cenários distintos. Com o objetivo de obter tais propriedades, enxames robóticos geralmente simulam o comportamento coletivo de insetos e animais, os quais apresentam intrincados mecanismos desenvolvidos pela evolução como soluções para vários problemas reais. Um requisito básico para a maioria dos enxames robóticos é a habilidade de navegar seguramente em ambientes compartilhados. Particularmente, um comportamento desejado é evitar a fusão de times diferentes que navegam em direções opostas. Esse é um exemplo de segregação, um fenômeno natural que é comumente observado na natureza. Vários sistemas biológicos se organizam de acordo com mecanismos baseados em comportamentos segregativos. Dentre esses, a segregação celular é de interesse particular pois desempenha um papel importante na formação de tecidos, órgãos e organismos vivos. Neste trabalho, um estudo sobre a segregação em enxames é apresentado e duas soluções são propostas para os problemas do agrupamento segregado e navegação segregada. A primeira abordagem se baseia na Hipótese da Adesão Diferencial, a qual afirma que células segregam naturalmente devido a diferenças de afinidade. Um controlador capaz de segregar enxames heterogêneos de acordo com as características de cada agente é introduzido, de modo que robôs semelhantes formem times homogêneos e robôs distintos fiquem segregados. Com relação ao segundo problema, um algoritmo baseado em abstrações hierárquicas, comportamentos de rebanho e obstáculos de velocidade é desenvolvido visando-se manter times de robôs segregados durante a navegação. Experimentos reais e simulados são analisados com o objetivo de estudar a viabilidade e eficiência dos métodos propostos. Os resultados mostram que as abordagens permitem um enxame de robôs heterogêneos segregar de maneira coerente e suave, sem a ocorrência de colisões.

Palavras-chave: robótica, enxames, segregação.

Abstract

Swarm robotics is the study of large multi-agent systems whose robots are relatively simple and have limited capabilities. These systems usually rely on properties such as robustness, flexibility, and scalability to fulfill complex tasks on distinct scenarios. In order to achieve these properties, robotic swarms generally simulate the collective behavior of insects and animals, which display intricate mechanisms shaped by evolution as solutions to many real-world problems. A basic requirement for most robotic swarms is the ability for safe navigation in shared environments. Particularly, a desired behavior is to avoid merging with different teams navigating in opposite directions. This is an example of segregation, a natural phenomenon which is commonly observed in nature. Several biological systems adopt self-sorting mechanisms based on segregative behaviors. Among these, cell segregation is of particular interest since it plays an important role in the formation of tissues, organs, and living organisms. In this work, we study segregation in swarm systems and propose solutions to two particular problems: segregated clustering and segregated navigation. We tackle the former by exploring the Differential Adhesion Hypothesis, which states that cells naturally segregate because of differences in affinity, and introduce a controller that can segregate heterogeneous swarms of robots according to the characteristics of each agent, such that similar robots form homogeneous teams and dissimilar robots are segregated. Regarding the latter problem, we present a distributed mechanism that combines concepts such as hierarchical abstractions, flocking behaviors, and velocity obstacles in order to maintain teams of robots segregated during navigation. We perform simulated and real experiments in order to study the feasibility and effectiveness of our methods. Results show that our approaches allow a swarm of multiple heterogeneous robots to segregate in a coherent and smooth fashion, without any interagent collisions.

Palavras-chave: robotics, swarms, segregation.

List of Figures

Contents

Chapter 1

Introduction

In this chapter, we introduce the context and motivation of this dissertation by presenting a brief overview of swarm robotics and its potential applications. We also define our research problem and discuss our main contributions to the subject of segregative behaviors in swarm systems. Finally, the chapter is closed with an outline of the organization of this document.

1.1 Swarm Robotics

Swarm robotics studies multi-agent systems consisting of a large number of relatively simple 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. Şahin [2005] defines swarm robotics as the study and implementation of methods that allow a large number of simple physically embodied agents to achieve a desired collective behavior which emerges from local interactions with the environment and among themselves. Moreover, he states that a robotic swarm should exhibit three system-level properties which are often observed in biological swarms:

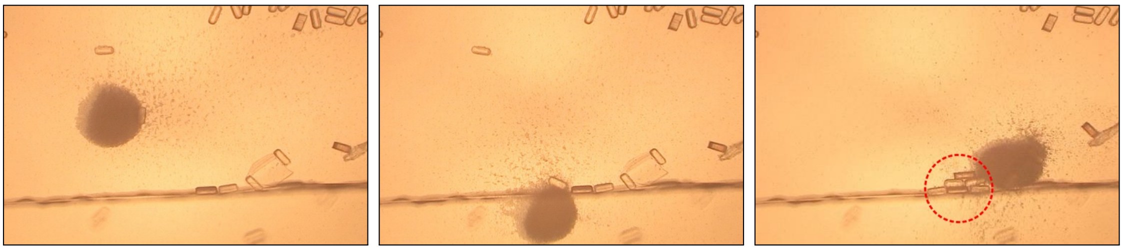
- **Robustness:** individuals should operate despite any agent malfunctions or external disturbances from the environment;
- **Flexibility:** the system should be able to cope with a wide range of different tasks and environments;
- **Scalability:** the swarm should support a large number of agents without significantly impacting its performance.

These properties provide advantages associated with the use of a swarm of simple robots over a few sophisticated ones, some of which are low-cost distributed sensing, lower chances of complete system failures, better workload distribution, and massive task parallelization.

Inspirations for swarm robotics usually come from observations of the collective behavior of insects and animals, which are exceptional examples of biological systems whose simple individuals can thrive when working together. For instance, the natural behavior of ant colonies, flocks of birds, and schools of fishes were used as foundations of many common approaches in swarm robotics, including *Reynold's flocking model* [Reynolds, 1987] and *Particle Swarm Optimization* [Kennedy and Eberhart, 1995]. In a broad sense, the study of emergent collective behaviors in self-organized systems is the main subject of *Swarm Intelligence*, which comprises the development of algorithms based on behavioral patterns of insect and animal societies [Bonabeau et al., 1999; Eberhart et al., 2001]. Emergent intelligence occurs when each individual has its own agenda, but the swarm as a whole seems highly organized towards a common goal. One example is the foraging behavior of ants, which is adapted according to the quality and spatial distribution of food sources scattered throughout the environment [Deneubourg et al., 1991]. In robotics, researchers can solve many problems by employing emergent behaviors such as aggregation, dispersion, foraging, self-assembly, cooperative transport, and pattern formation, most of which are achieved by solely relying on simple local interactions among individuals of the swarm. A thorough review of many swarm behaviors can be found in the works by Şahin [2005] and Barca and Sekercioglu [2013]. Furthermore, Parker [2008] gives a more general overview and classification of distributed intelligence in artificial systems.

Swarm systems have also a wide range of applications outside of robotics, e.g., digital animation [Reynolds, 1987], numerical optimization [Kennedy and Eberhart, 1995], web page classification [Holden and Freitas, 2004], crowd simulation [Narain et al., 2009], among others. In addition, civil engineers can use swarms to simulate a crowd evacuating from a building in case of emergency, and this procedure can lead to better design decisions that will minimize casualties in such a scenario [Thalmann and Musse, 2007]. Moreover, in order to reduce production costs, film directors often apply computer graphics and swarm technologies to create illusions of crowds, flocks, and herds. Notable examples include *Batman Returns* by Burton et al. [1992] and *The Lord of the Rings: The Fellowship of the Ring* by Jackson et al. [2001], in which swarms of bats and large-scale battle scenes were respectively computer-generated.

Besides robots and virtual agents, bacteria can be used as computer-controlled swarms to accomplish precise operations in micro-scale levels, since some types thereof can be steered by external forces such as those originated from magnetic fields. As a proof-of-concept, Martel and Mohammadi [2010] performed a complex micro-assembly experiment with magnetotactic bacteria, which were controlled by a set of coils capable of generating lines of magnetic fields in any desired direction. Figure 1.1 shows snapshots of an experiment from this work, in which a swarm of bacteria builds a micro-scale pyramid. Further research in this direction could lead swarm systems to deliver therapeutic agents into tumoral lesions inside the human body.



(a) Block pushing. (b) Block delivering. (c) Full pyramid.

Figure 1.1. Images from an optical microscope showing a swarm of approximately 5000 flagellated magnetotactic bacteria building a micro-scale pyramid. Source: [Martel and Mohammadi, 2010].

After this brief overview of swarm systems and their potential applications, it remains to introduce our problem. In this work, we explore a particular behavior of swarm systems known as segregation. We state our motivations and define the problem in the following section.

1.2 Motivation and Problem Definition

Most researchers in swarm robotics generally focus on homogeneous systems, in which all robots have the same physical characteristics. For instance, Şahin [2005] even includes in his definition of swarm robotics that agents should be rather homogeneous. However, several applications of multi-robot and swarm systems require the use of heterogeneous teams of agents in order to fulfill a given mission, as sometimes it is not possible to integrate all of the required sensing and actuation capabilities for the task in a single robot. This is the case of the *Swarmanoid* project [Dorigo et al., 2013], whose robots consist of three distinct types: foot-bots, a differential drive system with high mobility on rough terrain; hand-bots, an autonomous agent capable of manipulating objects as well as climbing vertical structures; and eye-bots, a flying robot designed to operate indoors (see Figure 1.2). These heterogeneous systems are especially useful on

(a) Foot-bots. (b) Hand-bots. (c) Eye-bot.

Figure 1.2. Robots from the Swarmanoid project. Source: [Dorigo et al., 2013].

cooperative assignments such as search and rescue, surveillance, perimeter protection, and cooperative transport.

In some cases, heterogeneous agents must be able to self-organize in a specific manner in order to complete their assigned tasks (e.g., see [Pimenta et al., 2008]). For instance, robots that gather distinct types of materials may need to form teams which can maximize the gathering of a particular resource. One strategy would be to sort agents according to their specialization, such that gatherers of similar materials stay in the same team. Afterwards, these groups can be deployed to different regions where a specific resource is abundant. We can say that such system shows a segregative behavior since the sorting process leads dissimilar agents into distinct teams.

Segregation is a particular sorting mechanism that is common in nature, being widely used by many individuals such as cells and animals to shape their populations into tissues as well as societies, respectively. This behavior has been extensively studied by biologists, but few robotics researchers have tried to simulate it on large swarm systems. Therefore, in this work, we study some basic mechanisms behind biological segregation and propose controllers in order to simulate this behavior. More specifically, we focus on two particular subproblems of segregation: clustering and navigation, which are defined below.

Segregated Clustering Problem (SEG-CLU). *Given a set of heterogeneous mobile robots, some of which share the same characteristics, sort all agents into homogeneous clusters such that each cluster contains robots with only the same characteristics.*

Segregated Navigation Problem (SEG-NAV). *Given a set of homogeneous robot clusters, each one having a particular goal position, navigate each cluster towards its respective goal while maintaining segregation among robots.*

Both problems refer to segregation in a spatial distribution sense, i.e., the traits of each robot directly influence its position in space. In this manner, similar agents form a cohesive cluster, and dissimilar ones are separated from each other. Traits can be either physical, such as having different sensors and actuators; or virtual, such as using broadcast identifiers to discriminate robots. Furthermore, we assume that every agent can perceive and recognize these characteristics by solely relying on onboard sensors, since perception and recognition are out of the scope of this work.

We tackle the SEG-CLU problem by developing a controller based on the *differential potential concept* [Kumar et al., 2010], an analogy for multi-agent systems of the biological mechanisms by which cells segregate. We also employ LaSalle's Invariance Principle in order to demonstrate convergence and present several simulated experiments in 2D and 3D spaces, which validate the proposed approach. With regards to the SEG-NAV problem, we investigate an algorithmic approach based on velocity obstacles and flocking behaviors in order to maintain segregation during robot navigation. In this method, 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. We present a series of simulated and real experiments in order to show the robustness of this method, and we analyze the results using a metric that measures the segregative behavior of the system along its execution.

1.3 Contributions

A summary of the main contributions of this dissertation to the swarm robotics community is as follows:

- We developed two robust and distributed methods, each of which is capable of solving the SEG-CLU and SEG-NAV problems, respectivelly, for multiple types of heterogeneous robots.

- We proposed a new metric in order to quantitatively define the spatial segregation of a heterogeneous system in an intuitive sense.

- We showed interesting properties of one of our controllers by employing a formal analysis based on Lyapunov Stability Theory [Lyapunov, 1992] and presenting several simulated experiments with hundreds of agents.

- We provided behavioral comparisons between one of our controllers and other previous approaches in order to show its advantages and disadvantages.

- Both of our approaches extend the Velocity Obstacle framework [Fiorini and Shillert, 1998] and the differential potential concept [Kumar et al., 2010].

During the development of this dissertation, we have produced some technical papers that have been submitted for publication. The following list provide references to these documents.

- Santos, V. and Chaimowicz, L. (2014). Cohesion and Segregation in Swarm Navigation. *Robotica*. Cambridge University Press.

- Santos, V., Pimenta, L., and Chaimowicz, L. (2014). Segregation of multiple heterogeneous units in a robotic swarm. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

- Santos, V., Campos, M., and Chaimowicz, L. (2012). On segregative behaviors using flocking and velocity obstacles. In *Proceedings of the 11th International Symposium on Distributed Autonomous Robotic Systems*.

1.4 Organization

We have organized this dissertation into five chapters, each of which is described below, with the exception of this introduction. Furthermore, we have separated our proposed solutions to the SEG-CLU and SEG-NAV problems into two chapters that contain the respective descriptions of our methods, experiments, and analyses.

- **Chapter 2 – Background:** discusses key concepts related to biological segregation and related work on coordinating multi-agent and swarm systems.

- **Chapter 3 – Segregated Clustering:** presents the methodology, analysis, and experiments related to the segregated clustering problem.

- **Chapter 4 – Segregated Navigation:** introduces the methodology, analysis, and experiments related to the segregated navigation problem.

- **Chapter 5 – Conclusion:** closes this dissertation with our conclusions and directions for future work.

Chapter 2

Background

In this chapter, we discuss some concepts related to biological segregation and present previous research that has attempted to simulate the phenomenon. Afterwards, we consider swarm segregation as a coordination problem and review some related work on the latter.

2.1 Biological Segregation Models

Segregation is a natural phenomenon which appears in several biological systems. For instance, ants sort their brood in annular patterns in which distinct broods tend to be placed at particular annuli [Franks and Sendova-Franks, 1992]; odors can impact the spatial distribution of several species of cockroaches, whose larvae prefer their own strain odor to that of another strain [Ame et al., 2004]; and the *Law of Segregation* from classic genetics explains trait inheritance as a process by which two genes separate from each other during gamete formation, and this process allows them to appear in different gametes of the offspring [Ridley, 2003]. Another example is cellular segregation, which is of central importance in embryogenesis, as the formation of many tissues requires an initial subdivision of cells into regions, each with specific characteristics that will allow particular cell types to be generated [Eduard and Wilkinson, 2012].

In order to explain the segregative behavior of cells, Steinberg [1963] postulated the *Differential Adhesion Hypothesis (DAH)*, which states that differences in cell adhesion generate mechanical forces that drive cellular segregation. In other words, a cell population experience stronger cohesive forces when among similar cells than when among dissimilar ones, and this imbalance is responsible for the segregative behavior [Eduard and Wilkinson, 2012]. Figure 2.1 shows two populations of cells that behave according to the DAH. Each cell type has been stained with distinct fluorescent membrane dyes in order to ease their discrimination. In this case, the DAH asserts that red cells envelope the green ones because the adhesion among the latter is stronger than among the former. As a matter of fact, cell adhesion was shown to be proportional to the protein expression levels of N-cadherins in a cell's surface [Foty and Steinberg, 2005], and since the two populations of Figure 2.1 actually express different levels of such proteins, these results serve as evidence of the DAH. Besides biological experiments, Agarwal [1995] successfully simulated segregation on a cellular automaton by employing the DAH. Figure 2.2 presents one of his experiments, in which white cells envelop black cells because of the same reasons as before. This simulation shows that Steinberg's model of cellular segregation can be implemented on multi-agent systems.

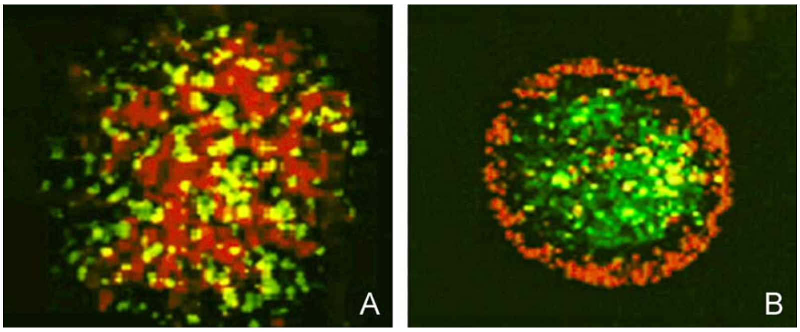


Figure 2.1. Cellular segregation of two distinct populations which express different levels of N-cadherins at their surfaces. Source: [Foty and Steinberg, 2005].

Robotics researchers have mostly focused on segregation as a mechanism by which robots sort a collection of objects. As an example, Deneubourg et al. [1991] developed a method that allows robots to sort similar objects into clusters by mimicking the foraging behavior of ants. However, some authors have specifically dealt with segregation of heterogeneous agents in the same sense as we do. In particular, Groß et al. [2009] discussed a motor schema that allows mobile robots to self-organize into annular structures. A distributed controller considers robots as having distinct virtual sizes and local interactions make "larger" robots move outwards. The procedure was inspired by the *Brazil Nut Effect*, a granular convection phenomenon by which a mixture of granular material subjected to vibrations leads its largest particles to the surface. This work was extended to consider real e-puck robots [Chen et al., 2012] as well. In spite of the interesting results, the controller requires that all robots share a common target in order to simulate the gravitational forces responsible for the granular convection, and this implies that a centralized broadcast or a consensus algorithm must be executed previously. Based on Steinberg's DAH, Kumar et al. [2010] proposed the *differential potential concept*, which asserts that agents should experience different magnitudes of

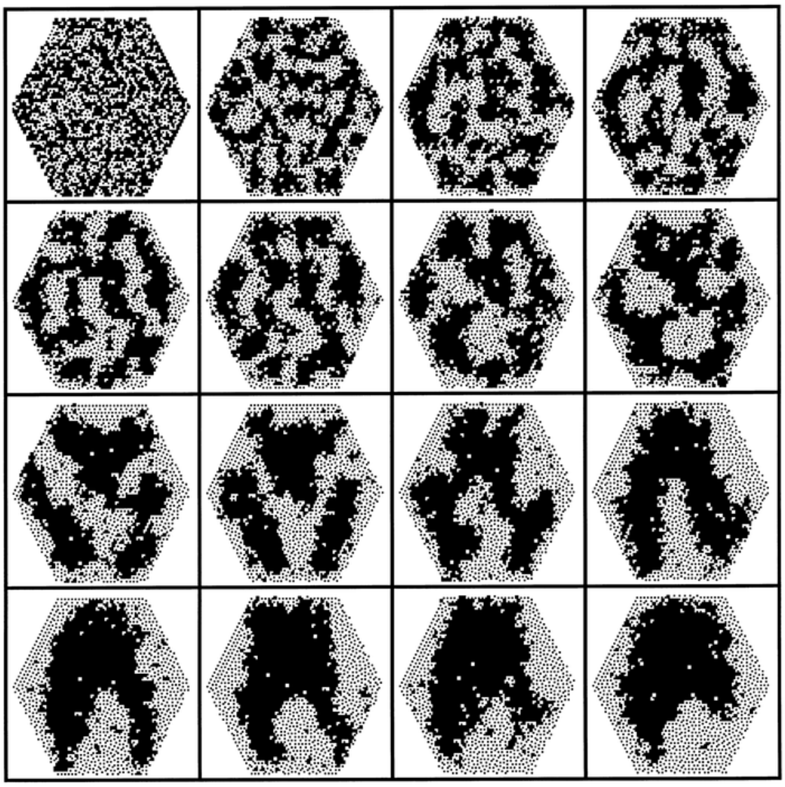


Figure 2.2. Discrete numerical simulation of cellular segregation based on the Differential Adhesion Hypothesis. Source: [Agarwal, 1995].

potential while interacting with agents of distinct types in order to achieve segregation. Stability analysis as well as convergence proofs were presented. Nevertheless, their approach is limited to only two types of robots and the use of of multiple types easily leads the system to local minima, where segregation does not occur, as we have seen through many experiments which we have performed with their controller. Finally, both of these works did not target the segregation of distinct groups during navigation, which is one of the problems that we focus. Moreover, we are interested in developing proper mechanisms that ensure segregation in the case of multiple robot types. We deal with these problems in Chapters 3 and 4, as accounted in Section 1.4.

2.2 Segregation as a Coordination Problem

Robots must coordinate themselves in order to achieve segregation, since at every time step any individual can influence the final position of other agents. Therefore, such behavior can be seen as a particular instance of the multi-robot coordination problem. As few works in robotics directly deal with the specific topic of this dissertation, we think it is important to review the research that has been done on coordinating multi-robot and swarm systems. We start our survey by discussing concepts related to path planning, since it is usual for coordination problems to be reduced into high-dimensional path planning problems.

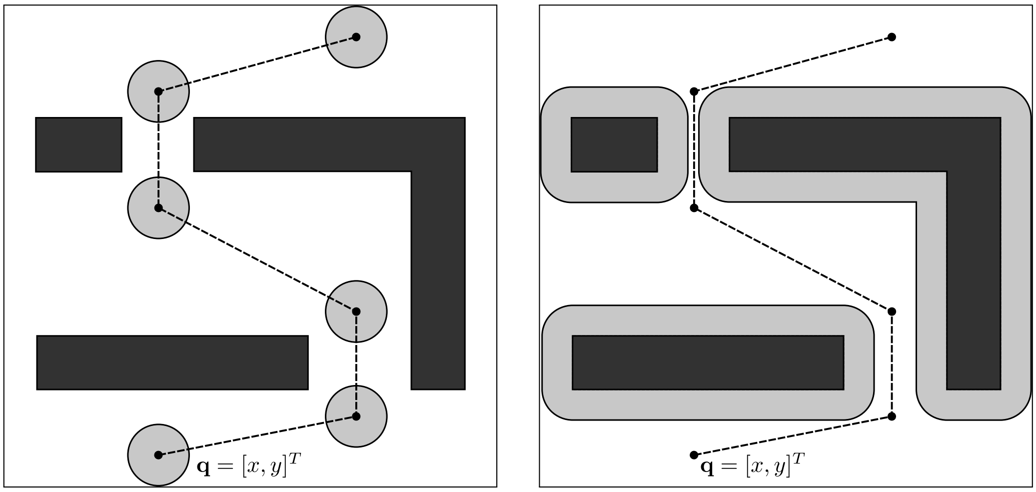
We classify related work into three main categories: centralized methods, which employ classical artificial intelligence and optimization techniques in order to coordinate the system as a whole; distributed methods, which specify individual behaviors that are responsible for achieving global coordination; and hybrid methods, which leverage the benefits over the disadvantages of the previous approaches. It is usual for centralized techniques to search for optimal solutions according to some criteria, such as the total navigation time, whereas decentralized methods often achieve coordination by compromising optimality over efficiency and scalability. In the middle of these two extremes, hybrid approaches aim to find a balance by employing one or more centralizing agents that help the underlying distributed system to achieve coordination.

2.2.1 Centralized Methods

Most centralized techniques achieve coordination in multi-robot scenarios by searching for a path in a composite space that comprehends all of the system's degrees of freedom. That is, they reduce coordination to a path planning problem, which consists in finding a collision-free path in an environment populated with obstacles. In its original formulation, known as *the piano mover's problem* [Reif, 1979], the task requires a piano to be safely moved throughout a furnished room.

The path planning problem was precisely defined and showed intractable during the 70s [Reif, 1979], and combinatorial solutions were developed during the 80s [Latombe, 1991]. However, these were restricted to systems with few degrees of freedom, which in turn hindered the use of multi-robot and swarm systems. The 90s and the following years have seen the development of sampling-based path planning [Kavraki et al., 1996; LaValle and Kuffner Jr., 2001; Karaman and Frazzoli, 2011], whose efficiency has allowed real industrial problems to be solved, even in areas outside of robotics such as automation, virtual prototyping, bioinformatics, and others. The books by Latombe [1991]; Choset et al. [2005]; LaValle [2006]; and de Berg et al. [2008] present a thorough review of path planning algorithms as well as their underlying data structures.

Most of these algorithms rely on the system's configuration space, a concept introduced in robotics by Lozano-Perez [1983] and Brooks [1983]. In order to illustrate this idea, we depict in Figure 2.3 the configuration space of a fully-actuated circular mobile robot navigating in a planar workspace. Notice that workspace obstacles expand when being mapped into the configuration space, whereas the whole robot is reduced to a point. The motivation behind this transformation is simple: instead of planning a path for a complex rigid body, the problem just needs to be solved for a single point [Choset et al., 2005]. For instance, in Figure 2.3, it is easy to see that a continuous collision-free path in configuration space represents a valid path in the workspace. Although both spaces have the same dimension in this case, it is common for them

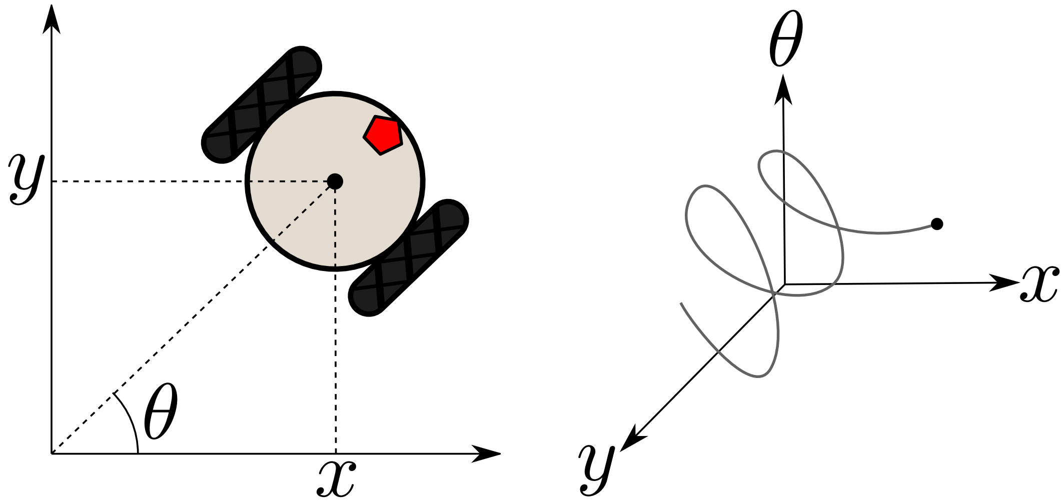


(a) Workspace $\mathcal{W} = \mathbb{R}^2$ (b) Configuration Space $\mathcal{C} = \mathbb{R}^2$

Figure 2.3. Workspace and Configuration Space of a fully-actuated circular mobile robot navigating in a plane.

to differ not only at their dimensionalities but also at their topologies. For example, consider a robot translating and rotating in a plane, as shown in Figure 2.4. It is clear that its workspace is $\mathbb{R}^2$, but its configuration space is in fact $SE(2)$ since we need two parameters for its position $(x, y) \in \mathbb{R}^2$ and one for its orientation $\theta \in S^1$ in order to completely specify all points of the robot.

When dealing with multi-robot systems, we can reduce the coordination problem into the path planning problem by considering a composite configuration space, which comprises all agents of the system. In this manner, finding a collision-free path over the Cartesian product of each robot's configuration space is a sufficient condition to coordinate them [Choset et al., 2005; LaValle, 2006]. Nevertheless, this approach is not scalable since this compound space exponentially grows with the number of robots. Due to this problem, researchers have developed alternatives whose main goal is to decouple the coordination from path planning. These include Kant and Zucker [1986] as well as O'Donnell and Lozano-Perez [1989], who presented a two-phase approach: in the first phase, they find a distinct path for all agents individually, i.e., without



(a) Workspace $\mathcal{W} = \mathbb{R}^2$ (b) Configuration Space $\mathcal{C} = SE(2)$

Figure 2.4. Workspace and Configuration Space of a underactuated circular mobile robot that navigates in a plane. Since $SE(2)$ is a manifold, we locally represent this space as $\mathbb{R}^3$.

taking into account other robots. Afterwards, in the second phase, authors resolve possible inter-robot collisions through velocity adjustments along these precomputed paths. More precisely, they parametrize all paths from the first stage and use these parameters in order to span the *path coordination space*. A path within the latter indicates the respective velocity modules that should be assigned at each point of a particular precomputed path during robot navigation. In another work, LaValle and Hutchinson [1998] discussed another variant of this approach, in which they employ a roadmap during the first phase instead of a single path per robot. Additionally, several authors including Erdmann and Lozano-Perez [1986]; Warren [1990]; Bennewitz et al. [2002]; as well as van den Berg and Overmars [2005] contributed to another alternative known as *prioritized planning*. In each iteration, a robot is sequentially selected according to a priority scheme, and a composite path is found by taking into consideration all robots of previous iterations, which are regarded as dynamic obstacles that move along predefined trajectories. The planner generally constructs a configuration space-time that represents all time-varying constraints imposed on the current robot by all static and dynamic obstacles [Erdmann and Lozano-Perez, 1986]. In a similar fashion, the technique by Saha and Isto [2006] searches for a path over the Cartesian product of the individual configuration space and the path coordination space that has been accumulated up to the current iteration. In the general case, these alternatives compromise completeness over efficiency, since the existence of a path in the system's composite configuration space does not guarantee a solution in the path coordination space [Saha and Isto, 2006]. Therefore, such methods should be employed only in specific coordination scenarios whose initial conditions allow solutions to be found.

It is usual to reduce path planning problems in configuration space to graph search problems either by cellular decomposition or by sampling configurations [Choset et al., 2005; de Berg et al., 2008]. Consequently, search methods such as Dijkstra's algorithm [Dijkstra, 1959], A* [Hart et al., 1968], or D* [Stentz, 1993] can be directly employed in order to find a feasible path. However, these methods are not suited for multi-robot coordination problems because of two particular reasons: the decomposition may suffer from the high dimensionality of the configuration space, and these search methods do not exploit the natural decoupling of robots that are well separated in the workspace. Wagner and Choset [2011] explored these facts and created a multi-robot search strategy called *subdimensional expansion*, which dynamically generates low dimensional search spaces embedded in the full configuration space of the system. Initially, each robot plans an individual optimal policy towards a goal configuration, and the planner expands the dimensionality of its search space as it finds inter-robot collisions among the optimal policies. At these critical configurations, the planner expands its search considering the Cartesian product of the tangent space of each robot's configuration space. In this manner, composite configuration spaces are explored only for robots that are indeed involved in collisions. This strategy effectively reduces the search space dimension, but it might underperform when dealing with swarm systems in highly-packed conditions, since chances of collisions are much higher. Finally, the subdimensional expansion strategy has received some attention in recent years and it has been coupled with sampling-based planners [Wagner et al., 2012] as well as other planning frameworks [Ferner et al., 2013].

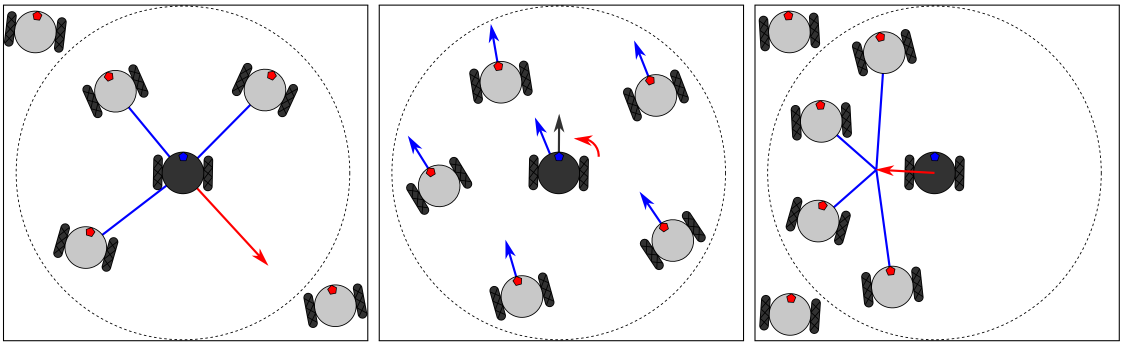
Many coordination problems in multi-robot systems do not require a specific robot to reach a particular goal. Such is the case of formation control, in which a permutation of two or more robots would not change the overall shape of their formation. Recently, researchers have been studying tailored solutions for these permutation-invariant systems. Kloder and Hutchinson [2006] modeled the formation as coefficients of a complex polynomial whose roots represent the robots' configurations. Their planner works in *formation space*, and robot trajectories are generated by tracking the roots of a interpolated complex polynomial. Moreover, Turpin et al. [2013a] found a permutation matrix that relates robots to goals, as well as straight-line and collision-free trajectories, through a functional optimization problem. They developed both centralized and decentralized algorithms that achieve optimal and suboptimal solutions, respectively, and extended their work in [Turpin et al., 2013b] to consider obstacles and not just straight-line trajectories. Nevertheless, their algorithm relies on constraints that hold only on initial conditions whose free regions have a homeomorphism to star-shaped sets, which may hinder its applicability in some scenarios.

All in all, we could employ one of the discussed methods in order to solve the SEG-CLU and SEG-NAV problems by selecting a compound configuration in which robots are segregated as the system's goal and expanding only states whose robots are segregated during the compound path planning, respectively. However, such solutions would not be proper because of the high dimensionality of the system's configuration space. Another disadvantage is that robots cannot be easily inserted or removed from the system, since the compound path must be recomputed at every insertion or removal. Essentially, the major drawback of centralized solutions is that they lack the required scalability to cope with swarms systems. On the other hand, distributed methods are well known to be scalable, and we delve into them in the next section.

2.2.2 Distributed Methods

Distributed approaches typically model the coordination of a robotic swarm through the specification of individual rules for each robot. Several distinct methodologies rely on this paradigm, including behavior based [Reynolds, 1987; Balch and Arkin, 1998], leader-follower [Tanner et al., 2004], and artificial potential functions [Leonard and Fiorelli, 2001; Olfati-Saber, 2006].

Reynolds [1987] 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 model comprises three simple steering behaviors that an agent applies based on its neighbors: *separation*, which avoids collisions; *alignment*, which steers the agents towards their average heading; and *cohesion*, which moves the agents towards their average position. In each iteration, the agent computes and weighs all rules according to user-specified parameters, and the result is applied as a steering input, as shown in Figure 2.5. Such interactions among agents can be modeled as a special case of the social potential field method [Reif and Wang, 1999], an extension of the classical artificial potential field technique [Khatib, 1985] that specifically deals with multi-agent systems. These works have been widely employed as foundations to several methodologies on the control of robotic swarms, such as behavior based [Balch and Arkin, 1998], leader-follower [Leonard and Fiorelli, 2001; Tanner et al., 2004], hierarchical abstractions [Belta and Kumar, 2004; Santos and Chaimowicz, 2011a], and hydrodynamic-based models [Pimenta et al., 2013]. Additionally, Olfati-Saber [2006] and Tanner et al. [2007] give a detailed analysis of the stability of flocking behaviors, as well as their robustness to topology changes in the neighborhood graph.



(a) Separation. (b) Alignment. (c) Cohesion.

Figure 2.5. Steering behaviors of Reynold's flocking model. Red arrows represent inputs and blue arrows describe data from neighboring agents.

Artificial potential fields [Khatib, 1985] direct robots as if they were particles moving in a vector field. A potential field is a differentiable map $U : \mathbb{R}^N \to \mathbb{R}$ whose image can be seen as energy, thus allowing its gradient ∇U to act as a force which can be feedbacked into the robot's controller. In other words, this means that one can devise a potential function such that robots act as charged particles which are attracted towards their goals while being repelled from nearby obstacles, as shown in Figure 2.6, and robots usually perform this procedure by following the negated gradient of the potential function. Khatib [1985] had originally conceived artificial potential fields as a trajectory planner, but later research has shown that the method is not oscillation-free and suffers from local minima, which is an intrinsic property that can arise from the combination of potentials, specially in unknown environments [Koren and Borenstein, 1991]. Figure 2.7 presents an example in which a robot is trapped because of a local minimum scenario. In this case, all forces acting on the robot have nullified each other, i.e., the gradient of the potential function has vanished. Solutions to the local minima problem include *navigation functions* [Rimon and Koditschek, 1992], which require configuration spaces that are diffeomorphic to star-shaped sets; and *harmonic functions* [Connolly, 1992; Pimenta et al., 2005], which generally need to discretize the configuration space in order to find a solution to Laplace's equation. Because of scalability issues, these preconditions generally hinder the direct application of such methods in swarm systems. Nevertheless, in spite of its theoretical limitations, potential fields have become a common tool in robotics, particularly as a local collision avoidance mechanism, due to its ease of implementation and its applicability to general classes of configuration spaces, such as multidimensional and non-Euclidean ones [Siegwart and Nourbakhsh, 2004; Choset et al., 2005].

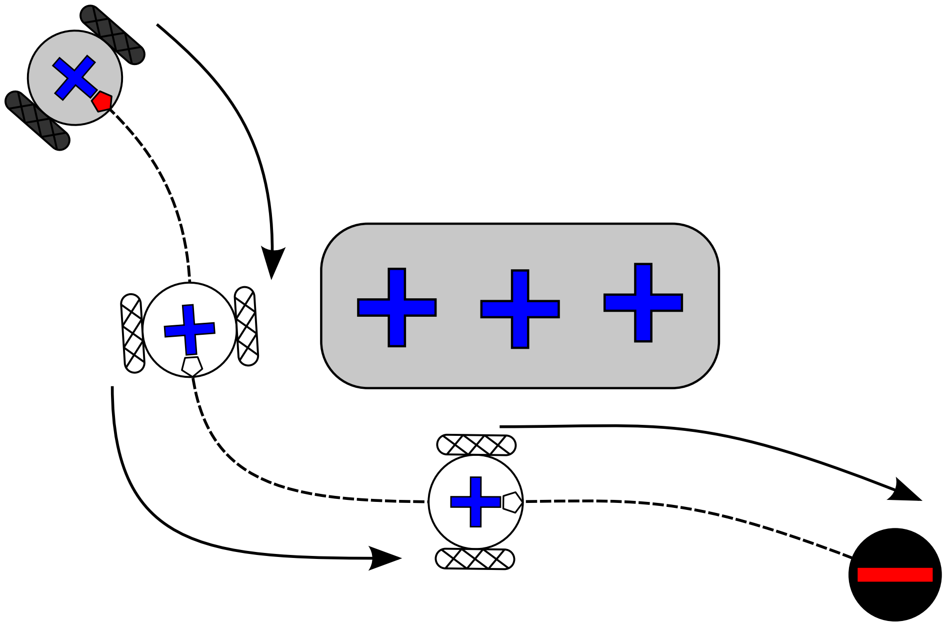


Figure 2.6. Navigation based on artificial potential fields. The robot is a positively charged particle that is attracted to its goal, whose charge is opposite, while being repelled by the gray obstacle.

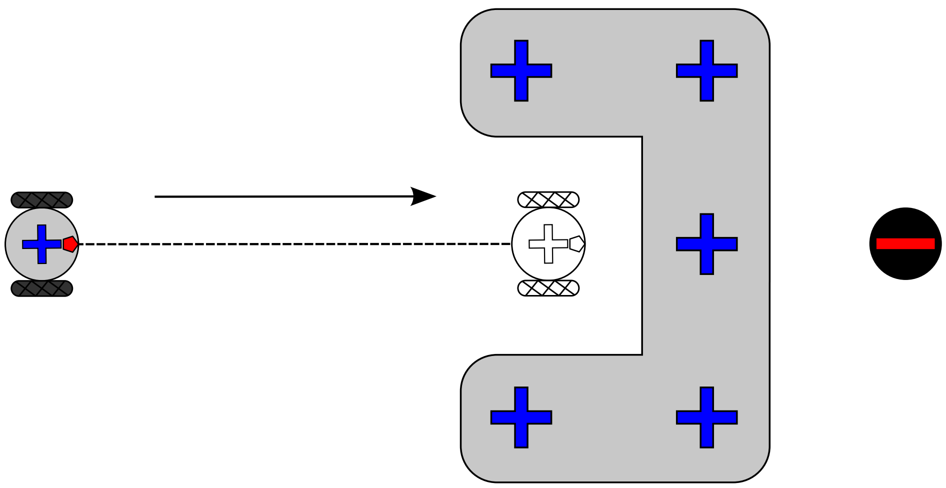


Figure 2.7. Local minimum scenario for navigation based on artificial potential fields. All forces acting on the robot sum to zero and thus it cannot reach its goal.

Several works in swarm robotics have focused on using artificial potential fields in conjunction with flocking rules in order to obtain specific coordinated behaviors, such as moving in formation [Balch and Hybinette, 2000], converging into shapes [Chaimowicz et al., 2005], area coverage [Howard et al., 2002], shepherding [Lien et al., 2004], and evasive maneuvers [Marcolino and Chaimowicz, 2009]. In a review of the area, Şahin [2005] discusses various swarm behaviors, many of which solely rely on potential fields.

Following a different approach, Desaraju and How [2011] proposed a distributed algorithm based on rapidly-exploring random trees [LaValle and Kuffner Jr., 2001], a well known sampling-based trajectory planner, in order to coordinate multi-robot systems. In this method, robots simulate the movement of each agent using waypoints and exchange bids containing their respective *potential path improvement*, a measure that reflects the expected improvement in path cost. A distributed auction determines the winning bid, thus allowing the winner to improve its current path, which is subsequently broadcast to all robots in the system. Despite having solved the particular coordination scenarios presented in the paper, the procedure requires all robots to know the control model of every agent in the system, so that their movements can be simulated, and since swarm robotics should commit to relatively simple robots, i.e., agents with low computational resources, such simulations would be impractical.

A distinct kind of low-level coordination consists in avoiding collisions among robots and obstacles in a reactive way, namely the obstacle avoidance problem. Several works explore the problem considering only one robot inside the environment, whose workspace may include dynamic obstacles with predefined trajectories [Borenstein and Koren, 1991; Fox et al., 1997; Fraichard and Asama, 2003]. If there are other robots in the environment, such algorithms typically assume that those agents are static objects or try to predict their respective velocities in each iteration. However, these approaches work only if robots navigate on low speeds with respect to the update frequency of the algorithm, and it is often the case that agents show oscillatory motions while avoiding each other. As indicated by van den Berg et al. [2008], such problems arise because the reciprocity of the avoidance behavior is oftentimes neglected. In other words, robots are not effectively coordinating themselves in a way that facilitates mutual avoidance. Techniques based on the concept of velocity obstacles [van den Berg et al., 2008, 2011; Snape et al., 2011] incorporate this cooperation into their methodologies.

A Velocity Obstacle [Fiorini and Shillert, 1998] is an extension of the Configuration Space Obstacle [Lozano-Perez, 1983] for a time-varying system. It defines the set of robot velocities that would result in a collision between the agent and an obstacle moving at a given velocity. Thus, the robot can perform avoidance maneuvers by selecting velocities that lie outside the velocity obstacle. The approach has been widely used and extended for single and multi-agent navigation [Abe and Yoshiki, 2001; Guy et al., 2009; Wilkie et al., 2009; van den Berg et al., 2011; Alonso-Mora et al., 2012, 2013], even when considering uncertainties in position, shape, and velocity of the obstacles [Fulgenzi et al., 2007; Snape et al., 2011; Hennes et al., 2012]. An important addition was the development of the Reciprocal Velocity Obstacle by van den Berg et al. [2008], who acknowledged that most works on collision avoidance had not taken into account the reciprocity that arises when obstacles are in fact other agents which can also react according to the robot's behavior. Recently, He and van den Berg [2013] presented another extension for the case in which a single agent should avoid a group as a whole. Despite relying on a virtual obstacle in a similar fashion as our *Virtual Group Velocity Obstacle* [Santos et al., 2012], which is introduced in Chapter 4, they do not focus on segregation since the method is restricted to a single agent instead of a whole group. Furthermore, [van den Berg et al., 2012] has recently developed Acceleration-Velocity Obstacles, showing that the Velocity Obstacle framework can be further extended in order to cope with dynamics.

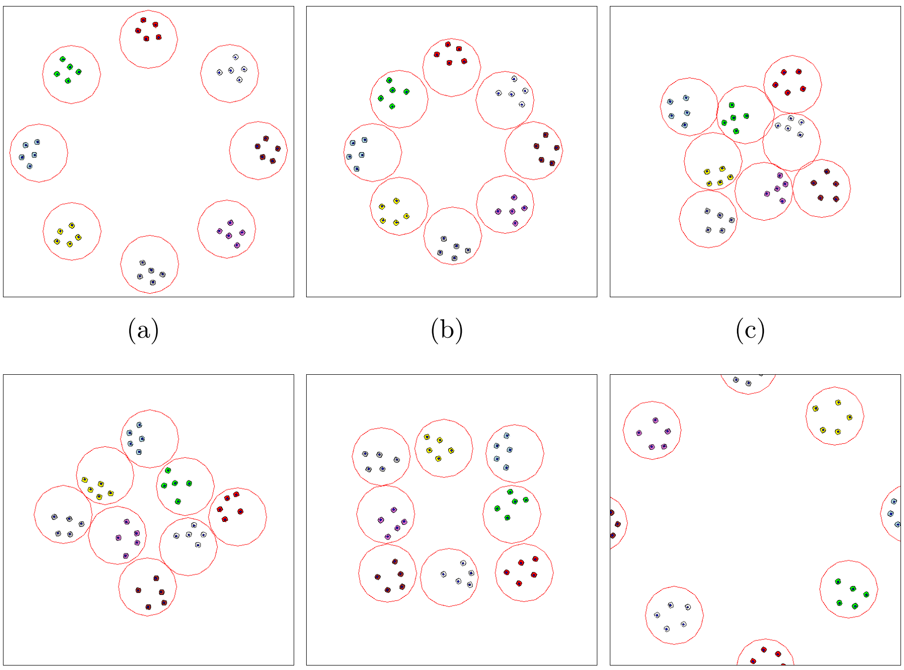
Few researchers have tried to use the Velocity Obstacle framework not only as a collision avoidance protocol but also as a coordination mechanism. For instance, Kimmel et al. [2012] devised algorithms that improve the coherence of a team of agents navigating in an environment with obstacles. In spite of the interesting results, it is still unclear how to directly use the framework to generate higher-level behaviors in a general setting. However, we believe that, because of their simplicity and theoretical foundations, further research in this direction might lead velocity obstacles to be as commonly employed as artificial potential fields in swarm systems.

In order to propose a solution to the SEG-CLU problem, we build upon the differential potential concept [Kumar et al., 2010], an analogy of Steinberg's DAH [Steinberg, 1963] that is based on artificial potential fields [Khatib, 1985]. Furthermore, we present a method that combines velocity obstacles [Fiorini and Shillert, 1998] with flocking rules [Reynolds, 1987] as an answer to the SEG-NAV problem. Regarding the latter, we introduce the *Virtual Group Velocity Obstacle*, a concept that has some of its foundations on hybrid methods.

2.2.3 Hybrid Methods

Hybrid coordination approaches exploit the benefits of both centralized and decentralized coordination schemes, while trying to minimize their disadvantages. We consider many of the following methods as semi-centralized, since it is usual to have a central unit computing a high-level plan that directly affects the behavior of the underlying distributed multi-robot system.

The *hierarchical abstraction paradigm* considers groups of agents as single entities in a hierarchical fashion. These entities are sometimes called virtual structures as they embody the pose and shape of a team of robots [Tan and Lewis, 1996]. In this case, steering control laws are applied to the virtual structure as a whole in order to maneuver the robotic swarm. We show in Figures 2.8 and 2.9 illustrative examples of this concept from our own prior work [Santos and Chaimowicz, 2011a,b], in which we chose ellipses as virtual structures that abstract the control equations of whole robotic groups. In other words, we needed to control only the parameters of all ellipses in order to coordinate an indefinite number of robots, and this has led to significant reductions in computational complexity when dealing with robotic swarms. For instance, in those works we minimized traffic congestions in crossroads scenarios (Figure 2.8) and navigated large teams of robots in environments with obstacles (Figure 2.9), without exploring configuration spaces whose dimensions grow exponentially with the number of robots. With regards to the latter figure [Santos and Chaimowicz, 2011b], we applied *Probabilistic Roadmaps* [Kavraki et al., 1996] in order to plan a path for the ellipse, and robots track the virtual structure through vector fields based on artificial potential fields [Khatib, 1985].



(a) (b) (c)

(d) (e) (f)

Figure 2.8. Example of traffic control in a crossroad scenario using the Hierarchical Abstraction Paradigm [Santos and Chaimowicz, 2011a].

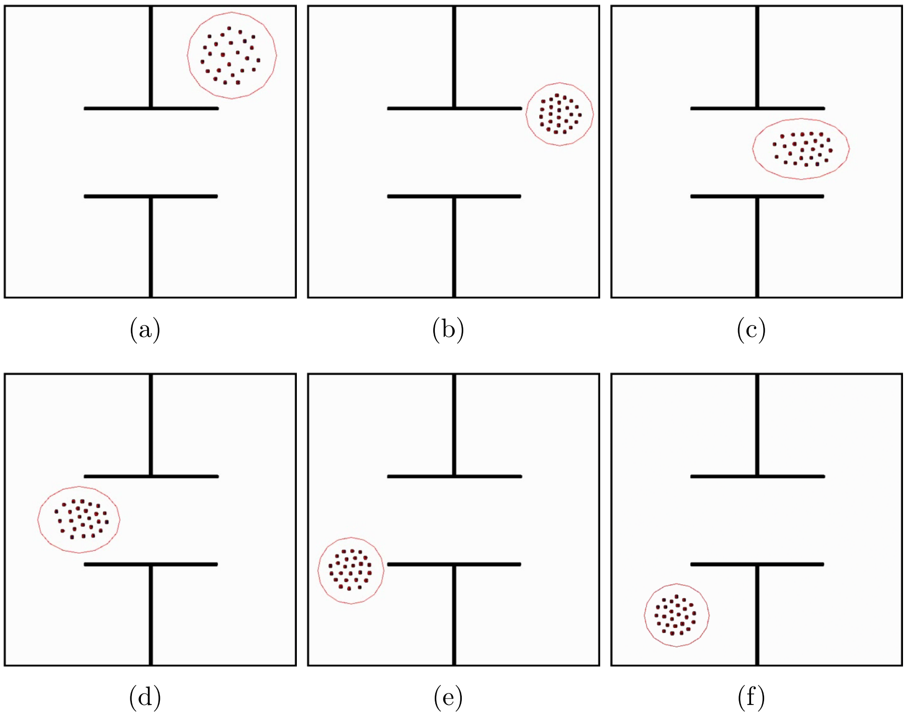


Figure 2.9. Example of motion planning using the Hierarchical Abstraction Paradigm [Santos and Chaimowicz, 2011b].

We can trace back work on virtual structures to papers by Tan and Lewis [1996] as well as Egerstedt and Hu [2001], who defined controllers that converge and maintain a team 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 [Barnes et al., 2009] and [Kamphuis and Overmars, 2004b] to group and control swarms of robots. In the latter, *Probabilistic Roadmaps* [Kavraki et al., 1996] were used to plan paths for the structure in environments with obstacles. However 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. Instead of considering a single deformable structure, some studies employed a set of structures to increase team cohesion and to simplify the path planning problem. For example, Li and Chou [2003] proposed a hierarchical sphere tree to control "crowds of robots", and in [Kamphuis and Overmars, 2004a] the path planned for a single agent is extended to a corridor using the clearance along the path. In this manner, it is possible to control a swarm that navigates through the corridor by changing its characteristics in a desired way. In another work, Belta and Kumar [2004] introduced a formal abstraction that allows decoupled control of the pose and shape of a team of robots. Their method 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. Michael et al. [2006, 2009] extended the work to account for three dimensional swarms and non-holonomic robots, and Hou et al. [2009] suggested a dynamic control model for similar abstractions. Moreover, Chaimowicz and Kumar [2007] discussed a distinct extension that tackles the cooperation between multiple unmanned aerial and ground vehicles, in which aerial agents estimate the configuration of ground robots and send control messages to the teams. The authors also studied merging and splitting behaviors, as sometimes such maneuvers are necessary for groups to overcome obstacles. However, interactions among teams with different goals were not addressed, and such scenarios can lead to traffic congestions in the environment when multiple teams try to navigate through the same region of space.

The traffic control problem is an important research topic, being characterized as a *resource conflict* problem [Cao et al., 1995]. In general, works in this area assume that robots are contained in a structured environment [Grossman, 1988; Kato et al., 1992], in which they navigate in delimited lanes that meet at intersections, usually where traffic control is performed. This can be accomplished by using a single manager agent [Dresner and Stone, 2005] or a more robust sensor network [Viswanath and Krishna, 2007]. As already cited, we employed a hybrid approach to traffic control in [Santos and Chaimowicz, 2011b], which in turn improved cohesion and maintained teams segregated. Even though this method contributes a solution to the SEG-NAV problem, it does not provide the required scalability since a centralizer agent is needed to compute the trajectory of all ellipses. Thus, in this dissertation we explore distributed approaches that achieve similar results by relying on flocking behaviors [Reynolds, 1987], velocity obstacles [Fiorini and Shillert, 1998], and inspirations from the hierarchical abstraction paradigm [Belta and Kumar, 2004].

Chapter 3

Segregated Clustering

In this chapter, we propose a solution to the SEG-CLU problem by extending the differential potential concept [Kumar et al., 2010] to deal with multiple robot types. Besides, we also introduce a new metric that defines segregation in a more convenient way, which can be easily verified. We perform a formal analysis on the properties of the proposed controller, and present simulated experiments in 2D and 3D environments.

3.1 Model and Definitions

We consider a set of fully actuated mobile agents whose dynamics are given by the double integrator

$$\dot{\mathbf{q}}_i = \mathbf{v}_i \text{ and } \dot{\mathbf{v}}_i = \mathbf{u}_i \qquad i \in \Upsilon = \{1, 2, \ldots, n\}, \tag{3.1}$$

in which $\mathbf{q}_i \in \mathbb{R}^p$, $\mathbf{v}_i \in \mathbb{R}^p$, and $\mathbf{u}_i \in \mathbb{R}^p$ denote the position, velocity, and control input of robot i, respectively. This set of mobile agents consists of different types of robots, which we represent by the partition $\tau = \{\tau_1, \tau_2, \ldots, \tau_m\}$, where each $\tau_k \subset \Upsilon$ contains all agents of type k. We assume that $\forall j, k : j \neq k \rightarrow \tau_j \cap \tau_k = \emptyset$ and $\forall j, k : |\tau_j| = |\tau_k|$, i.e., each robot is uniquely assigned to a single type and the type partition is fully balanced. Moreover, in this work, we focus on systems with $p = 2$ or $p = 3$.

Our objective is to synthesize a controller that can sort robots of different types into m distinct clusters in the workspace, such that each cluster contains agents of a single type only. The latter is a proper solution to the SEG-CLU problem, and a control system which solves this problem is said to display a segregative behavior.

3.2 Control Law

Given a population of n heterogeneous mobile robots with partition τ and dynamics specified by (3.1), we propose the following control law:

$$\mathbf{u}_i = -\sum_{j\neq i} \nabla_{\mathbf{q}_i} U_{ij}(\|\mathbf{q}_i - \mathbf{q}_j\|) - \sum_{j\neq i}(\mathbf{v}_i - \mathbf{v}_j), \tag{3.2}$$

in which $U_{ij}(\|\mathbf{q}_i - \mathbf{q}_j\|)$ is the artificial potential function that rules the interaction between agents i and j, $\|\mathbf{q}_i - \mathbf{q}_j\|$ is the Euclidean norm of the vector $\mathbf{q}_i - \mathbf{q}_j$, and $\nabla_{\mathbf{q}_i}$ is the gradient with respect to the coordinates of agent i. The first term represents the resultant force that acts on robot i given its interactions with all other agents, and the second term serves as damping, which causes robots to match their velocities. This kind of controller equation is a common approach for potential-based multi-agent systems [Leonard and Fiorelli, 2001; Olfati-Saber, 2006; Kumar et al., 2010].

The artificial potential field $U_{ij} : \mathbb{R} \to \mathbb{R}_{>0}$ is a function of the relative distance between a pair of agents that we express as

$$U_{ij}(\|\mathbf{q}_{ij}\|) = \alpha \left(\frac{1}{2}(\|\mathbf{q}_{ij}\| - d_{ij})^2 + \ln \|\mathbf{q}_{ij}\| + \frac{d_{ij}}{\|\mathbf{q}_{ij}\|} \right), \tag{3.3}$$

in which α is a scalar control gain, $\mathbf{q}_{ij}$ is a shortened form of writing $\mathbf{q}_i - \mathbf{q}_j$, i.e., $\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j$; and d_{ij} is a positive parameter that will be described later. The initial conditions and dynamics of the system exclude the situations where $\|\mathbf{q}_{ij}\| = 0$, in which (3.3) is undefined. As we will show later, if robots do not collide at the initial configuration then there will be no collisions through all time steps.

Although there are m distinct types of robots involved in the system, each agent classifies its neighbors as being either one of its own type or of a different type. This means that agents see the system through a binary filter which reduces possible robot interactions to only two kinds thereof: interactions among robots of the same type and among robots belonging to distinct types. Formally, we say that an agent i has a local type partition

$${}^{i}\tau = \{\tau_k, \Upsilon \setminus \tau_k\} \quad i \in \tau_k, \tag{3.4}$$

where $\tau_k \in \tau$, and $\Upsilon \setminus \tau_k$ represents the set difference.

In order to segregate robots, we apply the *differential potential* concept: when agents interact with other distinct agents, they experience different magnitudes of potential [Kumar et al., 2010]. We can accomplish this by defining the parameter d_{ij}

of (3.3) according to the local type partition ${}^i\tau$.

$$d_{ij}({}^i\tau) = \begin{cases} d_{AA}, & \text{if } i \in \tau_k \text{ and } j \in \tau_k \\ d_{AB}, & \text{if } i \in \tau_k \text{ and } j \notin \tau_k \end{cases}. \tag{3.5}$$

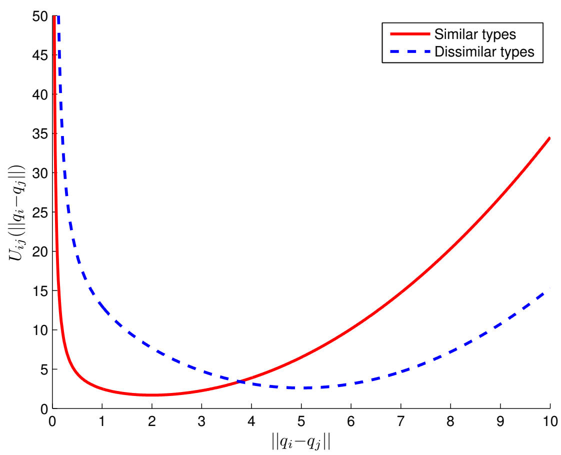
Equation (3.5) states that interactions among similar and dissimilar types of robots are ruled by d_{AA} and d_{AB}, respectively. Thus, the system exhibits a segregative behavior when we choose values for these parameters such that

$$0 < d_{AA} < d_{AB}. \tag{3.6}$$

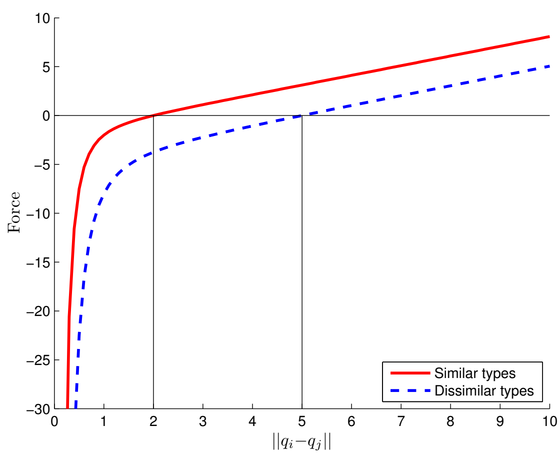
We show in Figure 3.1(a) a plot of the artificial potential function $U_{ij}(\|\mathbf{q}_{ij}\|)$, whose minimum is located at $\|\mathbf{q}_{ij}\| = d_{ij}$. Furthermore, we depict the interaction forces among a pair of robots in Figure 3.1(b), in which constraint (3.6) holds true. The latter plot actually represents the scalar part of the gradient

$$\nabla U_{ij}(\|\mathbf{q}_{ij}\|) = \alpha \left(\|\mathbf{q}_{ij}\| - d_{ij} + \frac{1}{\|\mathbf{q}_{ij}\|} - \frac{d_{ij}}{\|\mathbf{q}_{ij}\|^2} \right) \frac{\mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|}, \tag{3.7}$$

in which we ignore the normalized vector term. It is easy to see that, at any given distance, forces among agents of similar types are greater than those among different types. Therefore, our controller respects the Differential Adhesion Hypothesis [Steinberg, 1963], which states that cellular segregation occurs because there are stronger adhesive forces among similar cells than among dissimilar ones. In this sense, we can



(a) Interagent potential.



(b) Interagent force.

Figure 3.1. Plot of the artificial potential field $U_{ij}(\|\mathbf{q}_j - \mathbf{q}_i\|)$ and its underlying forces given $d_{AA} = 2$ and $d_{AB} = 5$.

say that attractive forces simulate the adhesion between cells, whereas repulsive forces prevent any inter-agent collisions.

3.3 Formal Analysis

This section presents a study on some properties of the multi-agent system when using the proposed control law. We employ Lyapunov Stability Theory and LaSalle's Invariance Principle in order to analyze convergence and the overall behavior of the swarm at the steady state. We start with the definition of the Lyapunov candidate function

$$V(\mathbf{q},\mathbf{v}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{v}^\intercal \mathbf{v}, \tag{3.8}$$

where $\mathbf{q} \in \mathbb{R}^{np}$ and $\mathbf{v} \in \mathbb{R}^{np}$ are stacked vectors whose components are the configurations and velocities of all robots, respectively, and $U(\mathbf{q}) : \mathbb{R}^{np} \to \mathbb{R}_{>0}$ is the collective artificial potential function, which we write as

$$\begin{aligned} U(\mathbf{q}) = &\frac{1}{2}\sum_{\tau_k \in \tau}\sum_{i \in \tau_k}\sum_{j \in \tau_k, j \neq i} U_{ij}(\|\mathbf{q}_i - \mathbf{q}_j\|) \\ &+ \frac{1}{2}\sum_{\tau_k \in \tau}\sum_{i \in \tau_k}\sum_{j \in \Upsilon \setminus \tau_k} U_{ij}(\|\mathbf{q}_i - \mathbf{q}_j\|). \end{aligned} \tag{3.9}$$

The first and second term of (3.9) are the sum of the pairwise potential between all pairs of similar and dissimilar robots, respectively. Thus, we can model the collective dynamics of the system by

$$\dot{\mathbf{q}} = \mathbf{v} \tag{3.10}$$

$$\dot{\mathbf{v}} = -\nabla U(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{v}, \tag{3.11}$$

in which $\hat{L}(\mathbf{q}) = L(\mathbf{q}) \otimes I_p$ is the Kronecker product of the system's Laplacian matrix $L(\mathbf{q})$ and the $p \times p$ identity matrix I_p (for a complete description, see [Olfati-Saber, 2006]). These definitions let us introduce the proposition below.

Proposition 1. *Assuming that the underlying adjacency graph of the system is complete at all times, for any initial condition that belongs to the level set $\Omega_C = \{(\mathbf{q},\mathbf{v}) \mid V(\mathbf{q},\mathbf{v}) \leq C\}$, with $C > 0$, a heterogeneous system with type partition τ on n mobile agents, whose dynamics and control laws are respectively given by (3.1) and (3.2), asymptotically converges to the largest invariant set in $\Omega_I = \{(\mathbf{q},\mathbf{v}) \in \Omega_C \mid \dot{V}(\mathbf{q},\mathbf{v}) = 0\}$, without any inter-agent collisions. At this largest* *invariant set, the velocity of each agent is bounded, all velocities match, and the system's collective potential reaches a local minimum.*

Proof. We aim to demonstrate that $\dot{V}(\mathbf{q}, \mathbf{v}) \leq 0$ in order to apply LaSalle's Invariance Principle to show convergence. To achieve this, we can differentiate $V(\mathbf{q}, \mathbf{v})$ with respect to time and then substitute (3.10) and (3.11) as follows

$$\begin{aligned} \dot{V}(\mathbf{q}, \mathbf{v}) &= \dot{\mathbf{q}}^\intercal \nabla U(\mathbf{q}) + \mathbf{v}^\intercal \dot{\mathbf{v}} \\ &= \mathbf{v}^\intercal \nabla U(\mathbf{q}) + \mathbf{v}^\intercal(-\nabla U(\mathbf{q}) - \hat{L}(\mathbf{q})\mathbf{v}) \\ &= -\mathbf{v}^\intercal \hat{L}(\mathbf{q})\mathbf{v} = -\frac{1}{2}\sum_i \sum_j \|\mathbf{v}_j - \mathbf{v}_i\|^2 \leq 0. \end{aligned} \tag{3.12}$$

The last step holds because the system's adjacency graph is complete [Olfati-Saber, 2006]. From LaSalle's Invariance Principle, all initial conditions that lie on Ω_C will lead the system to the largest invariant set in Ω_I, where $\dot{V}(\mathbf{q}, \mathbf{v}) = 0$. Therefore, this constraint together with (3.12) imply that all velocities match (i.e., $\forall i, j : \mathbf{v}_i = \mathbf{v}_j$) since this is the only case in which (3.12) can be equal to zero. Furthermore, by applying the constraint $V(\mathbf{q}, \mathbf{v}) \leq C$ into (3.8), we conclude that $\mathbf{v}^\intercal \mathbf{v} \leq 2C$, which leads to $\|\mathbf{v}\| \leq \sqrt{2C}$. Consequently, all individual velocities are bounded by $\sqrt{2C}$ as well. Matching velocities imply that inter-agent distances remain constant; hence, $\forall i, j : \dot{\mathbf{q}}_{ij} = \mathbf{0}$, and we have

$$\begin{aligned} \dot{U}(\mathbf{q}) = &\frac{1}{2}\sum_{\tau_k \in \tau} \sum_{i \in \tau_k} \sum_{j \in \tau_k, j \neq i} \dot{\mathbf{q}}_{ij}^\intercal \nabla_{\mathbf{q}_{ij}} U_{ij}(\|\mathbf{q}_{ij}\|) \\ &+ \frac{1}{2}\sum_{\tau_k \in \tau} \sum_{i \in \tau_k} \sum_{j \in \Upsilon \setminus \tau_k} \dot{\mathbf{q}}_{ij}^\intercal \nabla_{\mathbf{q}_{ij}} U_{ij}(\|\mathbf{q}_{ij}\|) = 0, \end{aligned} \tag{3.13}$$

which allows us to conclude that $U(\mathbf{q})$ is constant at the steady state. Moreover, matching velocities also imply that $\hat{L}(\mathbf{q})\mathbf{v} = \mathbf{0}$, which reduces (3.11) to

$$\dot{\mathbf{v}} = -\nabla U(\mathbf{q}). \tag{3.14}$$

Therefore, $\nabla U(\mathbf{q})$ must be the zero vector, as otherwise the collective potential would reach a lower value instead of being constant. This implies that the system has reached a local minimum and velocities must not change. Finally, assume that robots i and j collide (i.e., $\|\mathbf{q}_{ij}\| = 0$) at some moment in time. We can see by (3.3) and (3.9) that this would take $U(\mathbf{q}) \to \infty$, but this contradicts the fact that $V(\mathbf{q}, \mathbf{v}) \leq C$ and $\dot{V}(\mathbf{q}, \mathbf{v}) \leq 0$. Hence, no agent collides with each other. □

3.4 Experiments

In order to measure segregation among clusters quantitatively, we propose a metric that is based on the pairwise intersection area of their convex hulls:

$$M(\mathbf{q},\tau) = \sum_{\tau_k \in \tau} \sum_{\tau_l \in \tau, l \neq k} A\left(CH(\bigcup_{i \in \tau_k} \mathbf{q}_i) \bigcap CH(\bigcup_{j \in \tau_l} \mathbf{q}_j) \right), \qquad (3.15)$$

in which $A(Q)$ and $CH(Q)$ denote the area and the convex hull of set Q, respectively. We have chosen this metric because the convex hull can be used as a simple and well-defined shape representation of a cluster. This means that segregation occurs when there is no overlap among clusters. In other words, we say that the system is fully segregated when $M(\mathbf{q},\tau)$ approaches zero.

We executed a sequence of experiments to study the performance and feasibility of our proposed approach. We first present the results according to metric $M(\mathbf{q},\tau)$ and then display some snapshots of these simulations. Finally, we close the section with a discussion on the behavior of the system as well as on particular details of our method.

3.4.1 Simulations

We have performed extensive series of simulations in order to analyze our controller under metric $M(\mathbf{q},\tau)$. Each simulation consisted of 150 robots and a varying number of agent types. At the initial state, all velocities were set to zero, and robots were positioned according to a two-dimensional uniform distribution, which is independent of a robot's type. Additionally, we have set $d_{AA} = 2$ and $d_{AB} = 5$ for all experiments. We present in Figure 3.2 the mean and standard deviation of $M(\mathbf{q},\tau)$ among 100 experiments given these initial conditions. In all cases, both the mean and standard deviation approach zero as the number of iterations increase. Moreover, systems with less robot types tend to achieve segregation faster than those which have more types. This is expected, as given a fixed number of robots, a large number of types would result in few robots per cluster, which in turn would lower the magnitude of the resultant attraction force towards it.

We display in Figure 3.3 a series of snapshots of particular instances from our experiments. Through a visual inspection, we can see that similar robots quickly form clusters whose size grows with time as other agents join them. Furthermore, interesting geometrical patterns are organized at the stable state. We have also observed two particular behaviors which might be difficult to notice in the figures: large ensembles usually move to the outside of the main aggregate, the one which embodies all agents

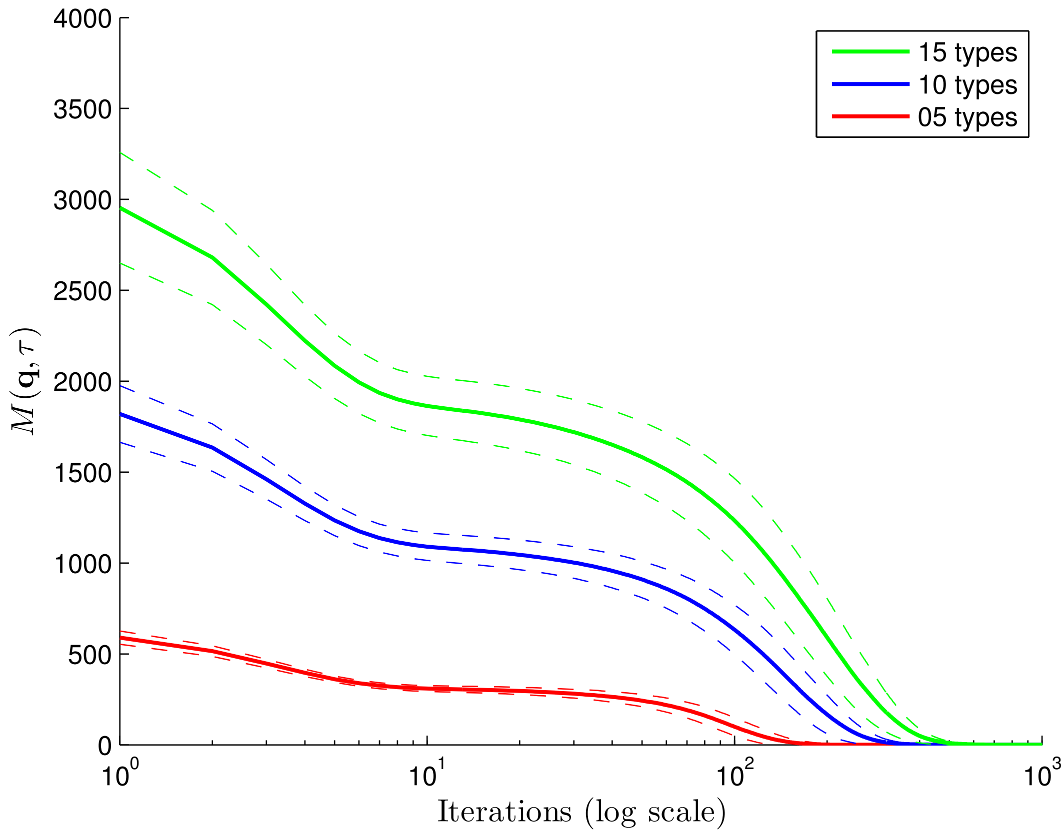
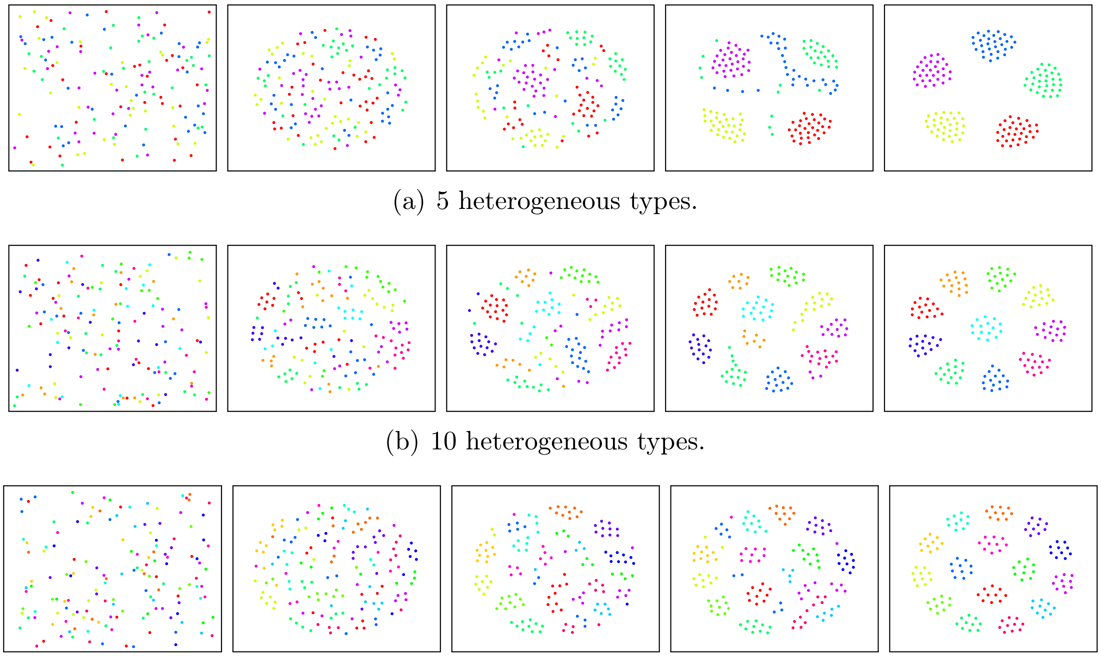


Figure 3.2. Mean intersection area of convex hulls for 100 experiments with a varying number of robot types. Dashed lines represent one standard deviation from the mean.

of the system, and adjacent dissimilar clusters form corridors which are used by agents of a third type to move at higher speeds. Both of these behaviors are compelling since they contribute to the opening of free spaces, whereby smaller clusters and lone robots can take advantage of the situation and form larger ensembles.

We have also executed simulations in 3D space. Figure 3.4 contains two images of the initial and final configurations from three experiments which comprised 150 robots and a varying number of agent types. Initial conditions were chosen exactly as in the 2D simulations. The controller was able to achieve segregation and robots have displayed the same overall behavior as of their 2D counterparts. Particularly, we have seen that it is easier for robots to form clusters in this scenario because the additional degree of freedom allows them to maneuver in new directions. Thus, in the 3D case, it is unusual to find lone robots wandering towards their cluster in later iterations, since larger clusters are aggregated more quickly.



(a) 5 heterogeneous types.

(b) 10 heterogeneous types.

(c) 15 heterogeneous types.

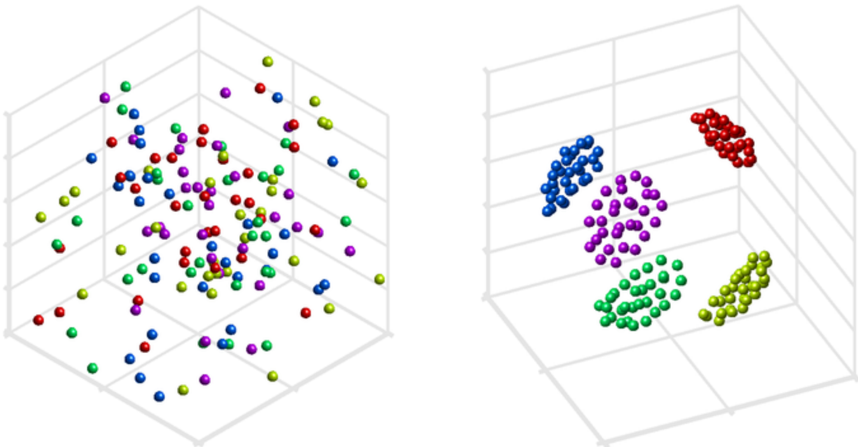
Figure 3.3. Snapshots of simulated executions with 150 robots for a varying number of heterogeneous types. In each sequence, the initial configuration is depicted on the left, whereas the final configuration is displayed on the right. Each robot type is represented by a different color.

3.4.2 Discussion

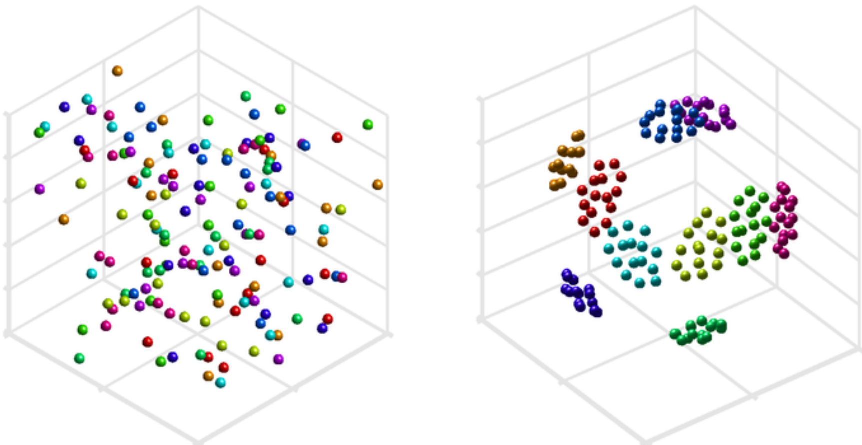
Desirable properties in swarm systems include scalability, flexibility, and robustness [Şahin, 2005]. These are especially important on applications in which robots may be inserted or removed from the system dynamically. One of the main advantages of our approach is that a robot does not need to know either how many agents or how many types exist in the system. This is due to the second case of (3.5), in which we write $j \notin \tau_k$ instead of $j \in \Upsilon \setminus \tau_k$ as the former explicitly states that robot i needs to recognize only agents that are similar to itself. Consequently, robots can be inserted or removed from the system at any time.

As can be seen in (3.2), our controller requires global perception capabilities. In other words, each robot must know the position and velocity of all agents. Thus, in spite of the robustness of our approach, this constraint may hinder the application of our method on real distributed systems, because most sensors have constrained capabilities that restrict robots to gather only local information.

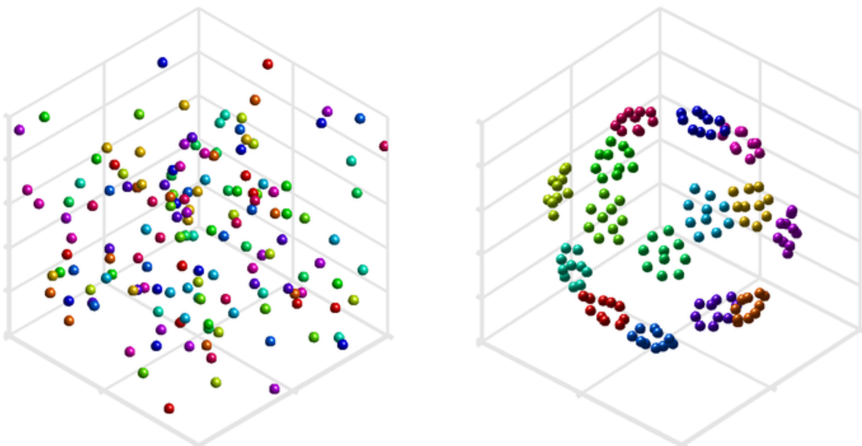
As mentioned, our controller is based on a previous work by Kumar et al. [2010], and both share the same advantages and disadvantages already cited. The main dif-



(a) 5 heterogeneous types.



(b) 10 heterogeneous types.



(c) 15 heterogeneous types.

Figure 3.4. Initial and final configurations of simulated experiments in 3D space with 150 robots and a varying number of heterogeneous types. We apply an orthographic projection and slightly rotate the stable state to better depict the separation among clusters.

ference between the methods lies at the definition of the potential functions, of which theirs can be written in our notation as

$$U'_{ij}(\|\mathbf{q}_{ij}\|) = \alpha \left(\ln \|\mathbf{q}_{ij}\| + \frac{d_{ij}}{\|\mathbf{q}_{ij}\|} \right). \tag{3.16}$$

We have noticed that the gradient of the above potential function could vanish on the free space among adjacent dissimilar clusters, and this is the reason why our initial experiments using their controller for multiple partitions had often reached undesirable local minima. Subtracting (3.16) from (3.3), we find

$$U_{ij}(\|\mathbf{q}_{ij}\|) - U'_{ij}(\|\mathbf{q}_{ij}\|) = \frac{\alpha}{2} \left(\|\mathbf{q}_{ij}\| - d_{ij}\right)^2, \tag{3.17}$$

which is a quadratic term that is responsible for the segregative behavior in the case of multiple partitions. By adding this term, we have actually biased the norm and direction of the gradient, allowing robots to keep moving towards their respective cluster.

Although we have limited our approach to balanced type partitions, we have also executed some simulations with unbalanced partitions. Among these, several experiments have reached local minima when the type partition was severely unbalanced. In these local minima, robots were not segregated in the sense of our proposed metric. This usually happens when robots cannot reach their cluster as the attractive forces towards it are weaker than those repelling them away from other agents. On the other hand, these types of local minima in experiments with balanced type partitions are not common. For instance, among the 100 experiments with 15 types presented in Figure 3.2, there was only one instance which did not segregate. However, given the same initial conditions, by choosing a larger value for $\frac{d_{AB}}{d_{AA}}$ the controller was able to achieve segregation. This result shows that the largest invariant set for a specific value of $\frac{d_{AB}}{d_{AA}}$ can be a state in which robots are not fully segregated. Thus, we think that these parameters should be chosen according to how many robots and types exist in the system, as larger numbers thereof may require wider corridors between dissimilar clusters so that the gradient of the potential function will not vanish.

We close this section with a proposition based on the evidences that we have gathered from our results. We present it as a conjecture since we have yet to prove it.

Conjecture 1. *Given a balanced type partition τ, and assuming that the underlying adjacency graph of the system is complete at all times, for any initial condition that belongs to the level set Ω_C, there exists a finite value r such that if $\frac{d_{AB}}{d_{AA}} > r$ then $M(\mathbf{q}, \tau) \to 0$ as the number of iterations approaches infinity.*

Chapter 4

Segregated Navigation

Given that robots are already segregated into homogeneous clusters, it would be interesting to deploy them to specific regions where a certain task must be performed, such as the gathering of a particular material. We would like robots to maintain segregation and to behave cohesively as a team during navigation. In this chapter, we propose a solution to problem SEG-NAV by employing an algorithmic approach using the velocity obstacle framework as a foundation. We start by briefly reviewing its core concepts and then we introduce our methodology.

4.1 Velocity Obstacles

As mentioned in Chapter 2, a Velocity Obstacle [Fiorini and Shillert, 1998] is the set of all velocities that will result in a collision between a robot and an obstacle moving at a given velocity. Thus, robots can perform avoidance maneuvers by simply selecting a velocity outside this set. To ensure a dynamically feasible maneuver, the dynamics of the robot and its actuator constraints are mapped into acceleration constraints, which can be transformed into the robot's velocity space. In the following paragraphs, we provide a more formal definition of the Velocity Obstacle and present an extension known as the Reciprocal Velocity Obstacle [van den Berg et al., 2008].

Given the same control model as in (3.1), let A and B be two circular robots moving on the Euclidean plane. The velocity obstacle $VO_B^A(\mathbf{v}_B)$ of B to A is the set of all velocities of A that will result in a collision with B at some instant in time if the latter maintains its current velocity $\mathbf{v}_B$ [Fiorini and Shillert, 1998]. To formally define $VO_B^A(\mathbf{v}_B)$, we specify $\lambda(\mathbf{a}, \mathbf{b})$ as a ray starting at point $\mathbf{a}$ heading in the direction $\mathbf{b}$

$$\lambda(\mathbf{a}, \mathbf{b}) = \{\mathbf{a} + t\mathbf{b} \mid t \in \mathbb{R}_{\geq 0}\} \tag{4.1}$$

and $B \oplus -A$ as the Minkowski sum of B and $-A$, in which $-A$ represents robot A reflected about its reference point

$$-A = \{-\mathbf{a} \mid \mathbf{a} \in A\} \tag{4.2}$$

$$A \oplus B = \{\mathbf{a} + \mathbf{b} \mid (\mathbf{a} \in A) \wedge (\mathbf{b} \in B)\}. \tag{4.3}$$

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

$$VO^A_B(\mathbf{v}_B) = \{\mathbf{v} \mid \lambda(\mathbf{q}_A, \mathbf{v} - \mathbf{v}_B) \cap (B \oplus -A) \neq \emptyset\}. \tag{4.4}$$

Figure 4.1(a) shows a diagram of $VO^A_B(\mathbf{v}_B)$. As can be seen, it is a cone with apex at $(\mathbf{v}_B)$ that infinitely grows as $\|\mathbf{v}\|$ takes large values. In order to compute (4.4), we first represent B in the configuration space of A by reducing the latter to a point and enlarging the former to the circle $B \oplus -A$. Afterwards, for a particular velocity $\mathbf{v}_A$, we test whether $\lambda(\mathbf{q}_A, \mathbf{v}_A - \mathbf{v}_B)$ intersects $B \oplus -A$. If this is the case, then $\mathbf{v}_A \in VO^A_B(\mathbf{v}_B)$, and robots will collide at some moment in time, which, if desired, can be calculated by solving the implicit ray-tracing operations induced by these intersection tests. On the other hand, if no intersections exist, then $\mathbf{v}_A \notin VO^A_B(\mathbf{v}_B)$, and no collisions will occur. Both of these claims assume that robots maintain their current velocities at all times. Thus, by construction, the set $VO^A_B(\mathbf{v}_B)$ partitions the velocity space of A into colliding and avoiding velocities [Fiorini and Shillert, 1998].

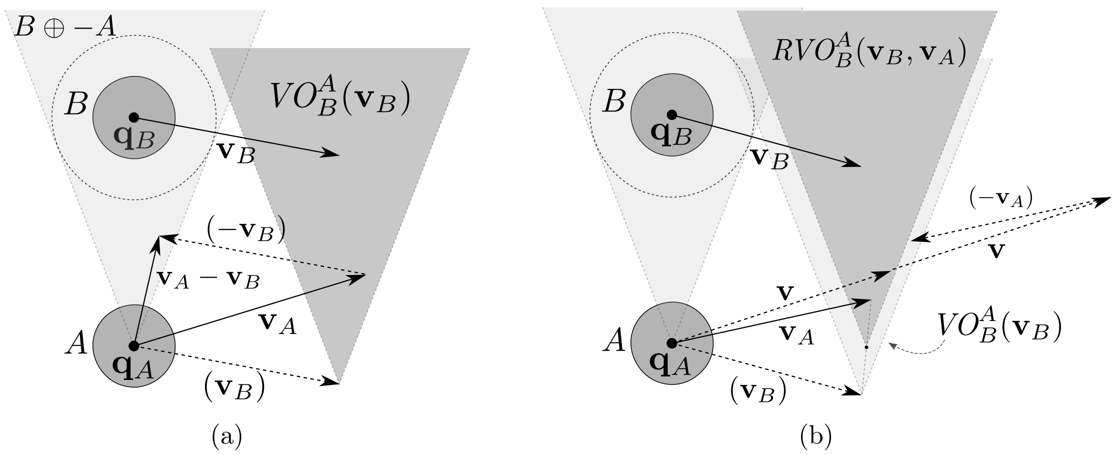


Figure 4.1. Diagrams of a Velocity Obstacle and a Reciprocal Velocity Obstacle. Adapted from [van den Berg et al., 2008].

The use of velocity obstacles can lead to oscillation issues when dealing with multi-robot systems. The problem is due to symmetry, which is a property that immediately follows from (4.4):

$$\mathbf{v}_A \in VO_B^A(\mathbf{v}_B) \iff \mathbf{v}_B \in VO_A^B(\mathbf{v}_A). \tag{4.5}$$

Given two robots A and B navigating with respective velocities $\mathbf{v}_A$ and $\mathbf{v}_B$ such that $\mathbf{v}_A \in VO_B^A(\mathbf{v}_B)$ and $\mathbf{v}_B \in VO_A^B(\mathbf{v}_A)$, it is clear that each one should select a new velocity $\mathbf{v}'_A$ and $\mathbf{v}'_B$ such that $\mathbf{v}'_A \notin VO_B^A(\mathbf{v}_B)$ and $\mathbf{v}'_B \notin VO_A^B(\mathbf{v}_A)$ in order to avoid a future collision. In such a scenario, we see by (4.5) that $\mathbf{v}_B \notin VO_A^B(\mathbf{v}'_A)$ and $\mathbf{v}_A \notin VO_B^A(\mathbf{v}'_B)$, which states that the old velocities are collision-free after the update. Therefore, if $\mathbf{v}_A$ and $\mathbf{v}_B$ directly point to the goal position of A and B, respectively, then it is reasonable for robots to select these velocities at the next iteration. This behavior clearly leads to oscillations when the whole process is repeated.

In order to overcome the oscillation problem, van den Berg et al. [2008] introduced the Reciprocal Velocity Obstacle (RVO), a simple extension that is able to smoothly navigate agents in a shared environment. The $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$ of B to A comprises all velocities of agent A that are the average between its current velocity $\mathbf{v}_A$ and a velocity within $VO_B^A(\mathbf{v}_B)$. Formally, we have

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v} \mid 2\mathbf{v} - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\}, \tag{4.6}$$

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 4.1(b). Assuming that B behaves reciprocally, if A selects the closest velocity to $\mathbf{v}_A$ outside the set $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$, then navigation is guaranteed to be collision- and oscillation-free. These are due to the properties below

$$\mathbf{v}_A + \mathbf{w} \notin RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) \iff \mathbf{v}_B - \mathbf{w} \notin RVO_A^B(\mathbf{v}_A, \mathbf{v}_B) \tag{4.7}$$

$$\mathbf{v}_A \in RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) \iff \mathbf{v}_A \in RVO_B^A(\mathbf{v}_B - \mathbf{w}, \mathbf{v}_A + \mathbf{w}). \tag{4.8}$$

Equation (4.7) asserts that if agents reciprocally select avoiding velocities, then both will be collision-free, and (4.8) states that old velocities cannot be collision-free after a reciprocal velocity update. These properties follow from the symmetry and translational invariance of the RVO (for a complete proof, see [van den Berg et al., 2008]). One mechanism that ensures a reciprocal velocity update is for both robots to select the closest velocity to their current one outside their respective velocity obstacles. More specifically, if $\mathbf{v}_A + \mathbf{w}$ is the closest velocity to $\mathbf{v}_A$ outside $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$, then, by symmetry, $\mathbf{v}_B - \mathbf{w}$ is the closest velocity to $\mathbf{v}_B$ outside $RVO_A^B(\mathbf{v}_A, \mathbf{v}_B)$. Hence, (4.7) and (4.8) are applicable, leading to a collision- and oscillation-free navigation.

4.2 Virtual Group Velocity Obstacle

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

As in Chapter 3, we assume a partition $\tau = \{\tau_1, \tau_2, \ldots, \tau_m\}$, in which $\tau_k \subset \Upsilon$ contains all agents of type k. However, only the constraint $\forall j,k: j \neq k \rightarrow \tau_j \cap \tau_k = \emptyset$ is necessary in this case, i.e., each robot must belong to only one partition, which is not required to be balanced. We also consider that mobile agents of different types are already segregated at the initial time step. Thus, the partition τ not only specifies which robot belongs to each type, but also that they form teams.

Let $\eta_i \subset \Upsilon$ be the neighborhood of robot i, which we define as the set of all robots inside an open ball of radius r centered at $\mathbf{q}_i$

$$\eta_i = \{j \mid (\|\mathbf{q}_i - \mathbf{q}_j\| < r) \wedge (i \neq j)\}, \tag{4.9}$$

and let $\Phi_k = \eta_i \cap \tau_k$ be the set of robots belonging to the partition τ_k that are within the neighborhood η_i, i.e., all agents of a particular team that are inside the sensing radius of robot i. Moreover, we declare $\mathbf{q}(\Phi_k)$ and $\mathbf{v}(\Phi_k)$ as the average position and average velocity of all robots belonging to Φ_k, respectively.

In order to achieve segregation, we block velocities that may lead a robot to merge with another team by introducing the *Virtual Group Velocity Obstacle* (VGVO). The VGVO is a simple concept: robot i senses the relative position and velocity of every robot j within the neighborhood η_i and builds the shape of each team of robots, with the exception of its own. In the workspace of robot i, these shapes are considered as virtual obstacles moving at the average velocity of their respective underlying robots. Thus, robot i can build a virtual velocity obstacle specifying all velocities that will lead to a collision with these shapes, assuming that they maintain their current average velocities. Formally, we define the VGVO of robot i induced by group Φ_k as

$$VGVO^i_{\Phi_k}(\mathbf{v}(\Phi_k)) = \{\mathbf{v} \mid \lambda(\mathbf{p}_i, \mathbf{v} - \mathbf{v}(\Phi_k)) \cap C(\mathbf{q}_i, \Phi_k) \neq \emptyset\} \tag{4.10}$$

$$C(\mathbf{q}_i, \Phi_k) = \text{Shape}(\bigcup_{j \in \Phi_k} R(\mathbf{q}_j)) \oplus -R(\mathbf{q}_i), \tag{4.11}$$

in which Shape(Q) is the shape of the set of points Q, such as the smallest enclosing disc, the convex hull, or the more general class of α-shapes [Edelsbrunner et al., 1983]; and $R(\mathbf{q}_i)$ denotes the set of points that represent robot i in its workspace. Figure 4.2 shows a diagram of the VGVO, in which the convex hull is used as the shape operator.

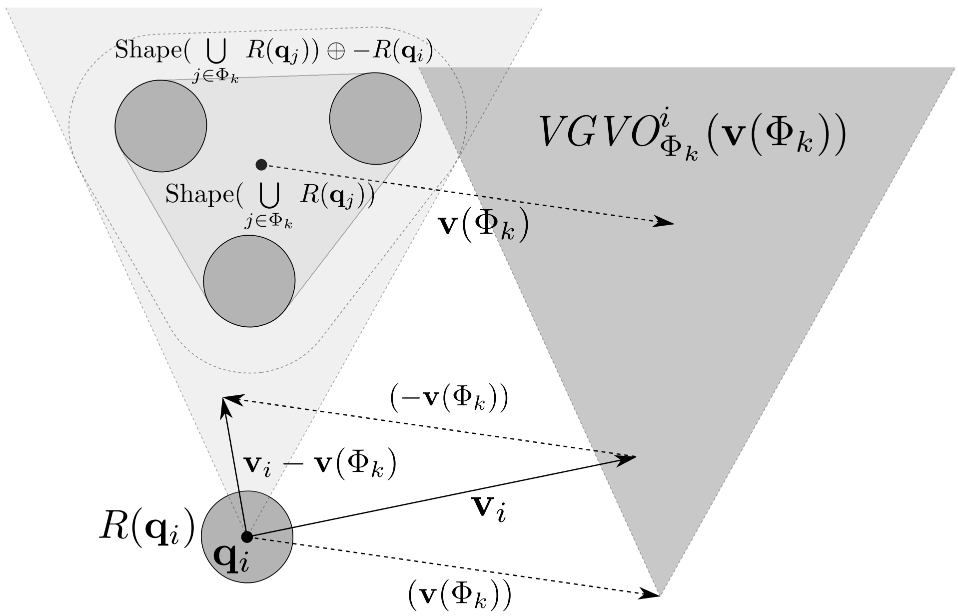


Figure 4.2. The Virtual Group Velocity Obstacle $VGVO^i_{\Phi_k}(\mathbf{v}(\Phi_k))$.

Equation (4.11) refers to the idea of the hierarchical abstraction paradigm (e.g., see Chapter 2), in which whole groups of agents are considered as single entities. Particularly, we abstract the set of agents Φ_k as a single entity that moves according to the average velocity of its underlying robots. During navigation, robots use the RVO in conjunction with the VGVO, which guarantees a collision-free navigation while maintaining the segregative behavior. Nevertheless, 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 update phase.

4.3 Velocity Update

An optimization problem must be solved in order to select inputs when dealing with Velocity Obstacles, and several distinct approaches have been developed [Fiorini and Shillert, 1998; van den Berg et al., 2008; Guy et al., 2009; van den Berg et al., 2011; Snape et al., 2011]. In this work, we achieve cohesion during velocity update by extending the algorithm from [van den Berg et al., 2008] to account for flocking rules. Basically, the method samples a set of admissible velocities, which are dynamically feasible, and selects the best one according to an utility function. Although other approaches have been developed to improve cohesion (e.g., [Kimmel et al., 2012]), flocking rules are widely employed in swarm systems, being interesting to couple them with the velocity obstacle framework.

In each iteration, robot i samples a set S of velocities using an uniform distribution from the admissible velocities

$$AV^i(\mathbf{v}_i) = \{\mathbf{v} \mid (\|\mathbf{v}\| < v_i^{max}) \wedge (\|\mathbf{v} - \mathbf{v}_i\| < a_i^{max}\Delta t)\}, \tag{4.12}$$

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 $\mathbf{v}_i$ given the robot's kinematic and dynamic constraints. In order to sample from $AV^i(\mathbf{v}_i)$, we actually sample from the input space of robot i and then transform samples into velocity space, since this makes it easier to deal with all constraints on the robot.

Let $\mathbf{v}_i^{\text{pref}}$ be the preferred velocity of robot i, such as the vector pointing at the robot's goal with magnitude bounded by the maximum allowed speed. Among the velocities in S, robot i should select a velocity outside all VGVOs and RVOs induced by agents in η_i, and this velocity should also drive the robot in the direction of $\mathbf{v}_i^{\text{pref}}$ while maintaining cohesion with its teammates. However, as the environment may become crowded to the point that no admissible velocities exist, the robot is allowed to select a velocity belonging to a velocity obstacle, but this choice is penalized according to the following function:

$$\mathbf{v}_i^{\text{flock}} = \mathbf{v}_i^{\text{pref}} + \alpha(\mathbf{v}(\Phi_k) - \mathbf{v}_i) + \beta(\mathbf{q}(\Phi_k) - \mathbf{q}_i) \tag{4.13}$$

$$P_i(\mathbf{v}) = \frac{\omega}{T_i(\mathbf{v})} + \|\mathbf{v}_i^{\text{flock}} - \mathbf{v}\|, \tag{4.14}$$

with $i \in \tau_k$. In the above equations, α weighs the alignment of the new velocity to the average velocity of teammates, β weighs the convergence of robot i to the centroid of its team, and $\omega \in \mathbb{R}_{\geq 0}$ regulates the avoidance behavior between aggressiveness and sluggishness. In other words, lower values of ω cause robots to favor flocking over collision avoidance, whereas higher values induce the opposite. Finally, $T_i(\mathbf{v})$ is the expected time to collision of robot i moving at velocity $\mathbf{v}$, which is computed by solving the ray-tracing equations induced by (4.6) and (4.10) for all agents in η_i. In the case of circular robots, these equations directly reduce to ray-circle and ray-segment intersection tests, which are trivially solvable. Thus, robot i selects the velocity $\mathbf{v}_i^{\text{new}}$ that minimizes the penalty function P_i over the sampled set $S \subseteq AV^i(\mathbf{v}_i)$.

$$\mathbf{v}_i^{\text{new}} = \operatorname*{argmin}_{\mathbf{v} \in S} P_i(\mathbf{v}) \tag{4.15}$$

Figure 4.3 exemplifies an iteration of the sampling-based velocity update.

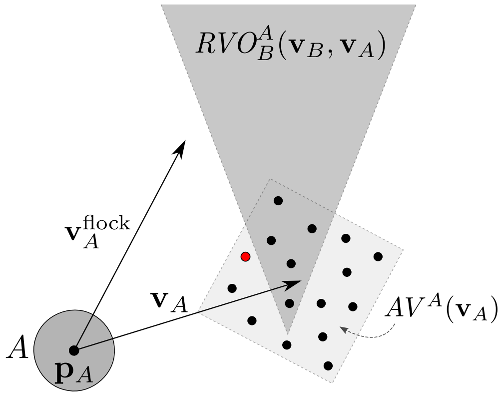


Figure 4.3. Sampling-based velocity update. Samples are represented by small circles. The red sample is chosen as it minimizes the penalty function.

Given two samples $\mathbf{v}$ and $\mathbf{v}'$, if both belong to any velocity obstacle, then the first term of (4.14) is well-defined, and, assuming that $\|\mathbf{v}_i^{\text{flock}} - \mathbf{v}\| = \|\mathbf{v}_i^{\text{flock}} - \mathbf{v}'\|$, the robot will select the safest velocity among the two, i.e., the one with a higher time to collision. On the other hand, if both are collision-free, then the first term of (4.14) tends to zero, which will make the robot select the closest velocity to $\mathbf{v}_i^{\text{flock}}$. In the general case, the value of ω balances the sampling-based velocity update between these two behaviors.

4.4 Experiments

In this section, we compare the hierarchical abstraction approach from [Santos and Chaimowicz, 2011a] to the proposed VGVO in terms of their segregative behavior as well as the time taken by each team 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 [Kumar et al., 2010]. We do not employ $M(\mathbf{q}, \tau)$ from (3.15) since the intersection area among convex hulls would be zero at every time step if the system were to remain segregated during navigation. Thus, the use of the latter metric would not bring any further insights besides knowing whether the system is segregated.

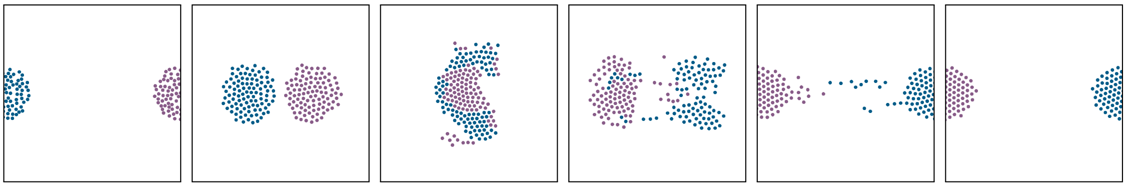
Additionally, we present experiments with two other methodologies for swarm navigation: basic attractive/repulsive potential fields [Khatib, 1985] and reciprocal velocity obstacles [van den Berg et al., 2008]. We selected these methods in order to show how the chosen metric from [Kumar et al., 2010] reflects the behavior of controllers that do not consider segregation. The basic artificial potential field approach consists of 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 [Choset et al., 2005]. With regards to the RVO algorithm, we have implemented it according to its description in Section 4.1, and the mechanism of Section 4.3 was used at each iteration to select collision-free velocities. Moreover, flocking behaviors were inhibited by setting constants α and β to zero.

4.4.1 Simulations

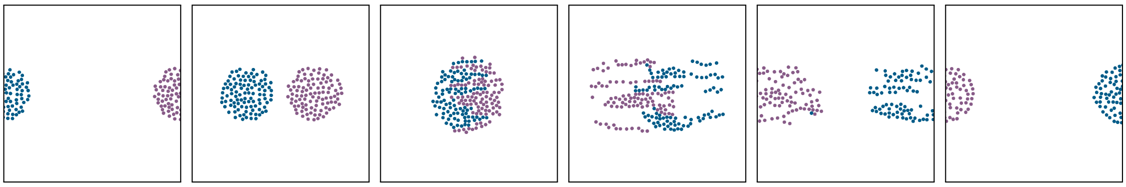
Each simulation consists of a crossroad scenario where robots are evenly partitioned into distinct teams. Initially, agents are randomly positioned according to a normal distribution into a circular area around their cluster's initial position. Afterwards, teams are commanded to swap their positions. 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. In order to properly reflect the mathematical definition of the VGVO, we have used α-shapes [Edelsbrunner et al., 1983] as a shape descriptor of each group, since they can describe concave as well as convex shapes. However, to completely define the VGVO, it is sufficient to look at two agents that maximize their radial distance from each other in the frame of reference of robot i. This can easily be seen in Figure 4.2, in which the VGVO would have the same aperture if the middle agent were not in the depicted group. Therefore, this feature can be used to optimize the implementation.

Figure 4.4 shows two groups of one hundred robots swapping their positions using all four presented methods. As can be seen, the hierarchical abstraction (Figure 4.4(c)) and the VGVO (Figure 4.4(d)) are capable of maintaining cohesion and segregation. On the other hand, neither potential fields (Figure 4.4(a)) nor RVOs (Figure 4.4(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 repulsive/attractive potential field leads groups to directly crash into each other, whereas the hierarchical abstraction prevents agents from mingling with different groups because of the avoidance behavior of its virtual structure.

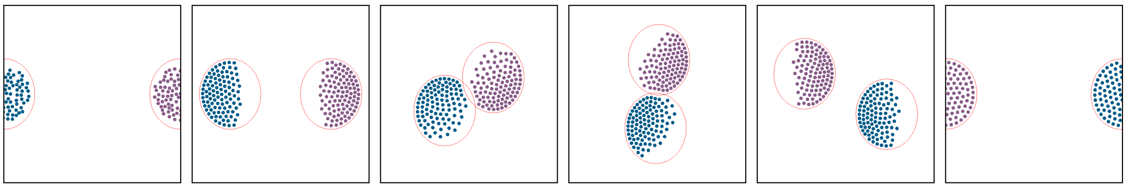
As already mentioned, Kumar et al. [2010] proposed a formal metric for measuring segregation between two teams of agents. In their work, two teams τ_A and τ_B are said to be segregated if the average distance among robots in the same team is less than the average distance among robots in different teams. In other words, the following



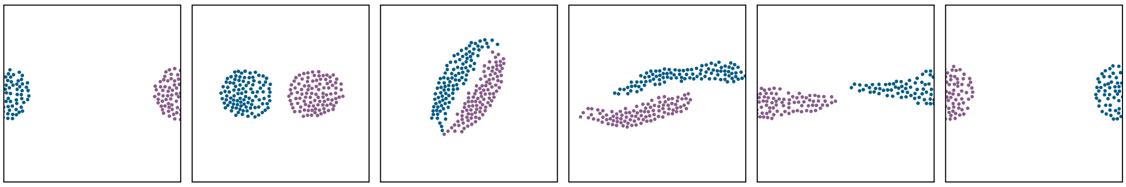
(a) Attractive/Repulsive Artificial Potential Field.



(b) Reciprocal Velocity Obstacle.



(c) Hierarchical Abstraction.



(d) Virtual Group Velocity Obstacle.

Figure 4.4. Behavioral comparison among controllers with two hundred robots evenly distributed into two groups using local sensing.

restriction must hold at all times

$$(d_{\text{avg}}^{AA} < d_{\text{avg}}^{AB}) \wedge (d_{\text{avg}}^{BB} < d_{\text{avg}}^{AB}), \tag{4.16}$$

in which d_{avg}^{AB} is the average distance among agents of teams τ_A and τ_B.

In Figure 4.5, we depict these average distances for the latter simulations. As can be seen in 4.5(c) and 4.5(d), both the hierarchical abstraction and the VGVO have successfully achieved the segregative property in the sense of constraint (4.16). The remaining simulations show the behavior of the metric when segregation is not achieved. For instance, in Figures 4.5(a) and 4.5(b), the constraint is violated since we observe intersection points between the curves d_{avg}^{AB} and d_{avg}^{BB}.

Another important information that we can extract from Figure 4.5 is the total amount of time required to complete the task. This can be measured at the point when

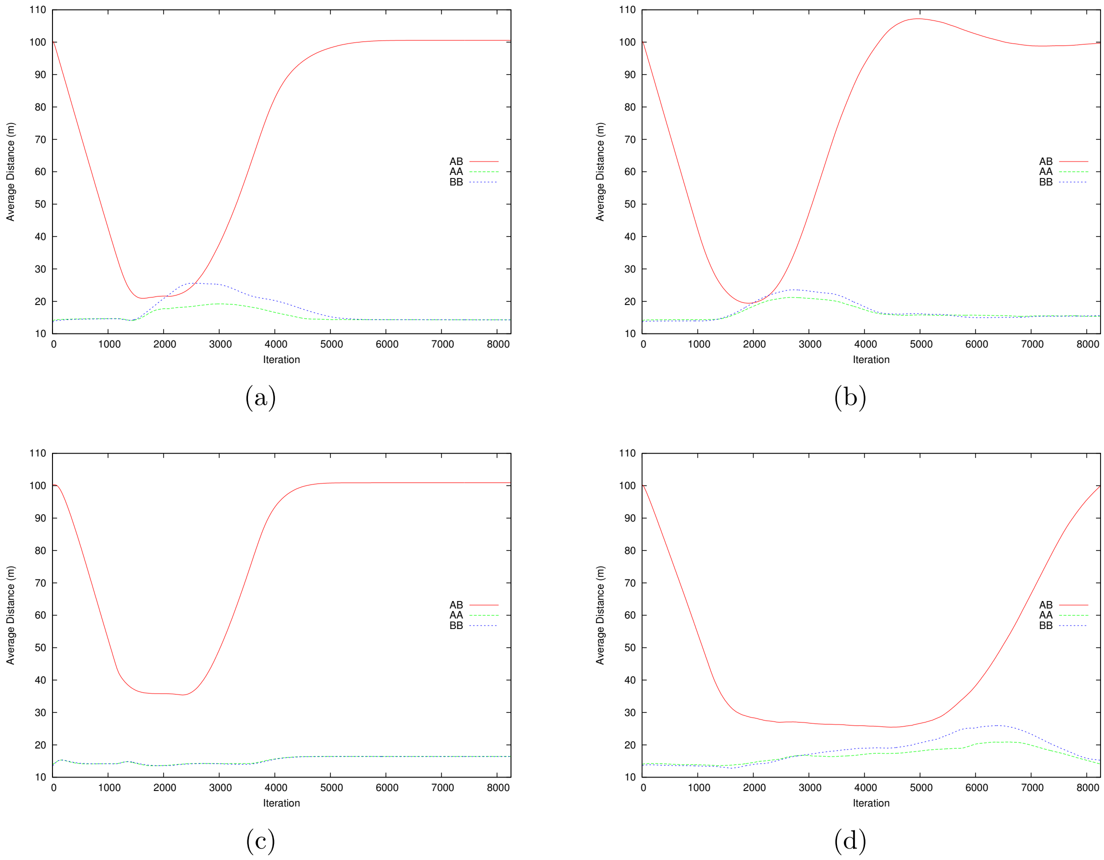
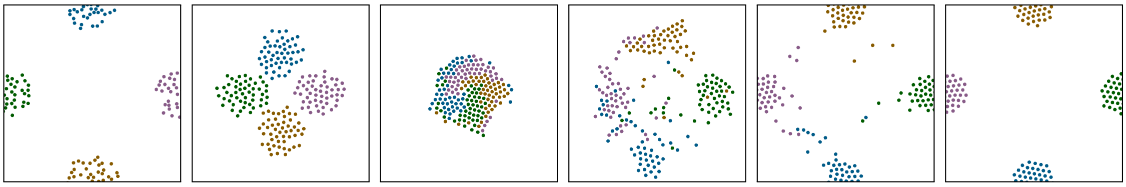
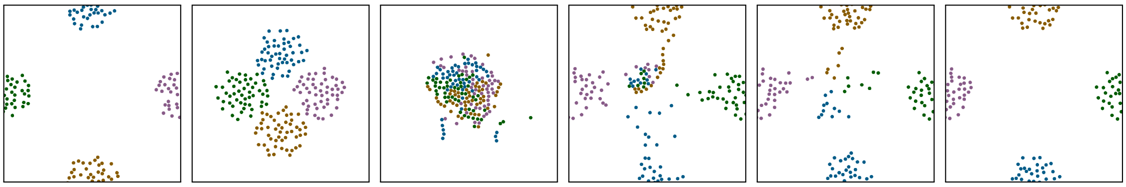


Figure 4.5. Segregative behavior analysis for two hundred robots evenly distributed into two groups that swap positions. (a) Attractive/Repulsive Artificial Potential Field. (b) Reciprocal Velocity Obstacle. (c) Hierarchical Abstraction. (d) Virtual Group Velocity Obstacle.

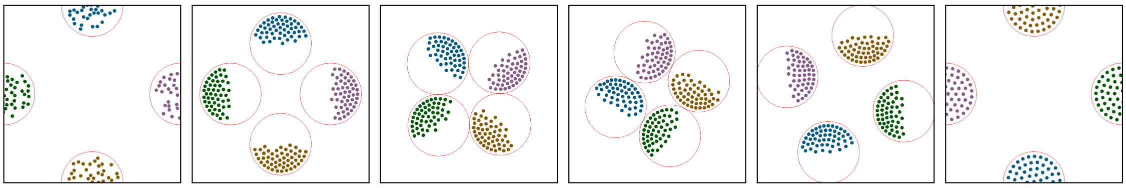
curve d_{avg}^{AB} returns to its initial value, which means that both groups have swapped their average positions. Instead of requiring that all particular goals are reached, we consider the swap between average positions as a requirement for the task because this metric emphasizes the group behavior over the individual behavior. Therefore, we can see that the RVO is the fastest approach, followed, in order, by the hierarchical abstraction, artificial potential fields, and the VGVO. The performance loss of the VGVO is mainly due to reciprocal dances [van den Berg et al., 2008], i.e., both teams cannot agree on which side to pass each other. The avoidance maneuver eventually takes place when robots reach a consensus, but this usually requires both teams to stretch into elongated shapes, as shown in Figure 4.4(d), which in turn can retard the completion of the task. The efficiency of the VGVO can be improved by introducing social rules, such as the one used in the method of Figure 4.4(c), in which ellipses always turn counterclockwise; or by biasing the velocity update towards one side of the VGVO, in a manner similar



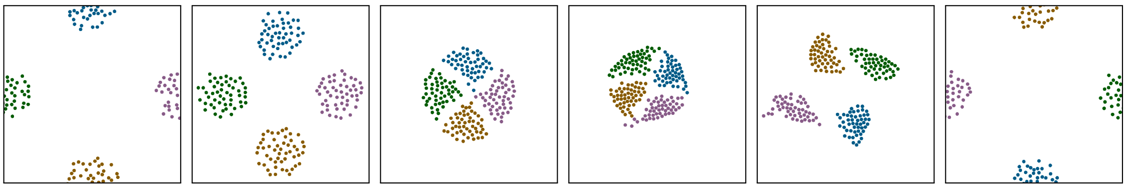
(a) Attractive/Repulsive Artificial Potential Field.



(b) Reciprocal Velocity Obstacle.



(c) Hierarchical Abstraction.



(d) Virtual Group Velocity Obstacle.

Figure 4.6. Behavioral comparison among controllers with two hundred robots evenly distributed into four groups using local sensing.

to the work by Snape et al. [2011]. Furthermore, the second and third terms of (4.13) also contribute to the efficiency loss because robots always select the safest velocity to maintain the flock. In congested scenarios, this results in slower speeds for the whole flock when compared to the RVO, which does not consider flocking behaviors.

We complement these results with another set of simulations in a similar scenario, but having two hundreds robots partitioned into four teams. Figure 4.6 illustrates these experiments. In Figures 4.6(a) and 4.6(b), robots form a large heterogeneous cluster in the center of the environment, which slowly dissipates while they reach their target positions. Moreover, a symmetrical avoidance behavior was achieved in the experiment of Figure 4.6(c) because we have used circles as the shapes of the virtual structures. Both simulations of Figures 4.6(c) and 4.6(d) have achieved cohesion and segregation in the sense of (4.16). We do not show the average distance plots for these experiments since the combination of all curves per team results in a cluttered graph. Nevertheless, results were similar to the ones obtained in Figure 4.5, i.e., both the hierarchical abstraction and the VGVO achieved segregation, and robots took a longer time to finish the task using the latter than the former.

4.4.2 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 [Mondada et al., 2009] (e.g., see Figure 4.7), which are small-sized differential robots equipped with a ring of 8 IR sensors for proximity sensing and a set of LEDs for displaying status. A bluetooth wireless interface allows local communication among robots and also with a remote computer. We controlled these robots through Player [Gerkey et al., 2003], a well-known framework for robot simulation and programming.

Figure 4.7. Twelve e-puck robots used in the experiments.

In order to estimate the configuration of all robots, we used a swarm localization architecture developed by Garcia et al. [2007], which is based on fiducial markers and overhead cameras, as shown in Figure 4.8. In this framework, one or more computers process the captured images and determine the position of all robots in a common frame of reference. Afterwards, control inputs are calculated according to our algorithm and broadcasted to the swarm through Bluetooth connections. Additionally, we implemented a virtual sensor to detect neighboring agents because of the limitations on the e-puck's IR sensors. Furthermore, to account for nonholonomic constraints, we transformed input velocities according to the approach presented in [Luca et al., 2000].

Figure 4.9 shows snapshots from executions of the VGVO approach. 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 4.5: The average distance between robots in

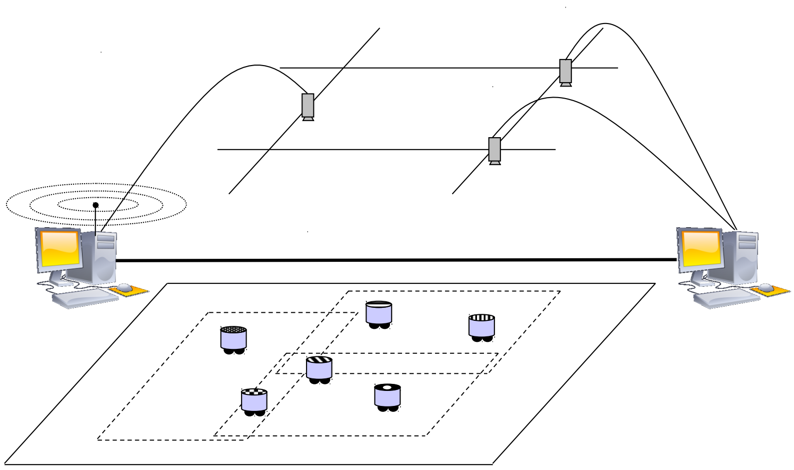


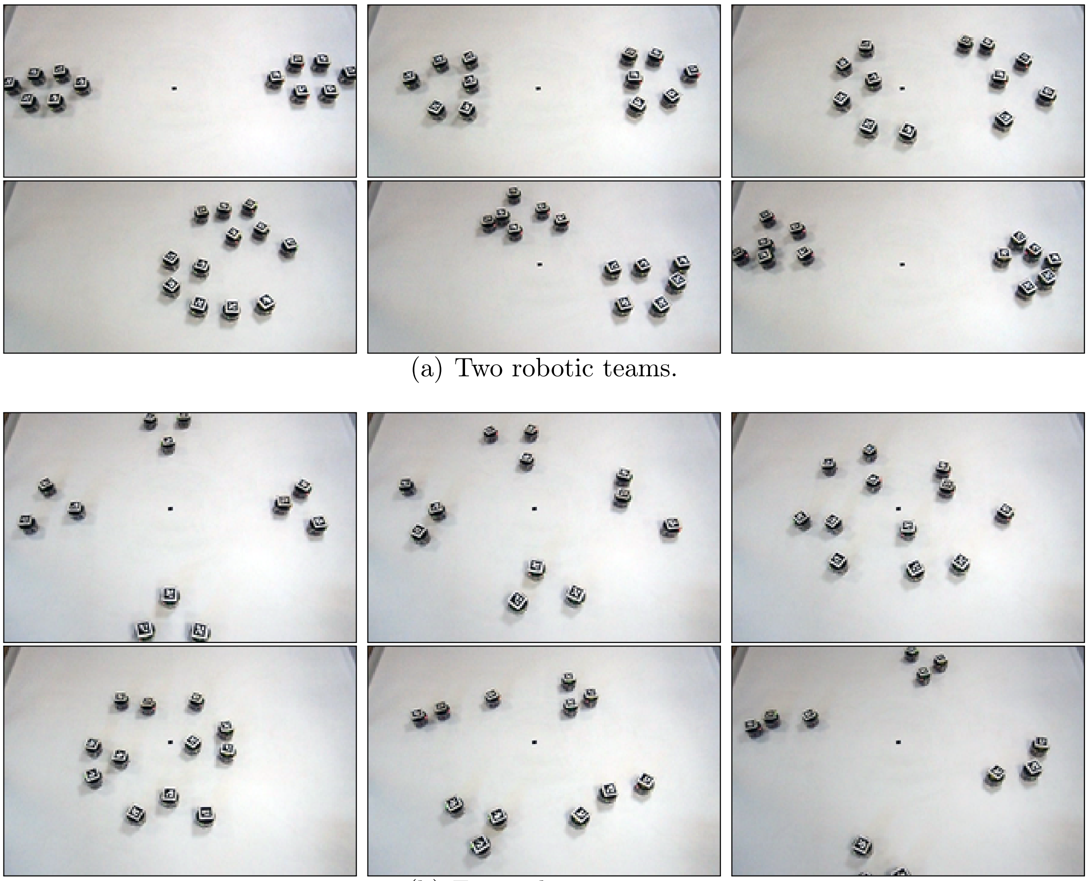
Figure 4.8. Schematic diagram of the architecture used in the experiments. Source: [Garcia et al., 2007].

the same team is always less than the average distance among robots in different teams. 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.

4.4.3 Discussion

The Velocity Obstacle framework is known for allowing high-speed navigation in multi-robot scenarios, but robots tend to prioritize slower speeds when using our approach. This result also influences the tuning process of all constants, since their values can impact the efficiency of the system. For example, given a high value for α in (4.13), robots will quickly align their velocities to the average velocity of their neighbors, which can easily lead to overshoot goal positions as well as increase the chances of collisions with single robots moving at high speeds. Similarly, a high value for β may lead agents into tightly aggregated groups, which makes robots prefer slower speeds because most higher speeds will be inside some velocity obstacle.

We can see evidence of these discussed problems in Figure 4.5(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 4.5(d), but it is not shown because of the scale on the axes. In both algorithms, this issue arises because there is no damping over the robot's acceleration towards its goal. Moreover, the velocity matching term of (4.13) actually worsens the problem, and it is easy to see that agents may leave their goal while trying to match their velocities. Therefore, parameters α and



(a) Two robotic teams.

(b) Four robotic teams.

Figure 4.9. Real execution of the VGVO algorithm with different team sizes.

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

Since the VGVO uses average velocities in its definition, it does not necessarily guarantee segregation. For instance, if a single robot moving at a high speed passed another robot of a different team, then the latter could enter the team of the former. However, it is unusual for this to take place, because the flocking behavior does not enable particular robots to move at such speeds. Actually, agents tend to drive nearing the average speed of their neighbors. Another problem is that average velocities can lead to loss of information, i.e., they might not indicate the correct direction of movement for a group of agents. Thus, it may be interesting to build the VGVOs of some subsets in Φ_k as well. For instance, subsets of two or three can be considered in conjunction with the whole set, and these will block more velocities than the VGVO of Φ_k alone, increasing safety with regards to segregation.

Chapter 5

Conclusion

In this dissertation, we studied segregative behaviors in swarm systems and proposed distinct methods that allow robotic swarms to segregate in the context of clustering and navigation. Motivations behind this research included the spreading interest in heterogeneous multi-robot architectures and the recurrent use of segregation as a sorting mechanism in nature. More specifically, we explored two particular problems: *segregated clustering*, in which agents must be sorted into homogeneous clusters according to some characteristic; and *segregated navigation*, in which teams of robots must navigate towards their goal region while maintaining segregation among different teams.

Regarding the segregated clustering problem, we proposed a controller that sorts a system consisting of multiple heterogeneous mobile robots into homogeneous clusters, such that similar agents are segregated from dissimilar ones. Our approach extended the differential potential concept to multiple types of robots. In this framework, agents experience different magnitudes of potential when interacting with dissimilar agents, in a manner analogous to the *Differential Adhesion Hypothesis* from cellular biology. Furthermore, we presented stability analyses and several experiments in 2D and 3D scenarios. The results in simulation show agreement with the formal analysis of the controller and indicate that it is robust in terms of segregation. However, we are unable to determine whether the largest invariant set corresponds to states in which segregation always holds true. Despite these good results, there are still limitations that must be resolved. For instance, assumptions such as global sensing and balanced partitions are generally not practical in real scenarios, but these constraints may be loosened by employing other potential functions that consider local sensing and asymmetry in their formulations. All in all, we expect that further study in this sense can provide improved solutions that are applicable in a wider variety of instances.

With respect to the segregated navigation problem, we introduced the *Virtual*

Group Velocity Obstacle concept, a set of velocities that can lead a robot to merge with a team of agents. Particularly, the VGVO resembles ideas from the hierarchical abstraction paradigm, in which robotic teams are considered as single virtual entities. We also maintained cohesion among agents by coupling the velocity obstacle framework with flocking behaviors. In our method, the robot's preferred velocity is biased to account for flocking rules, and velocities are updated according to a sampling-based procedure. We performed several experiments in simulated and real scenarios, whose results demonstrated the effectiveness of the proposed approach. Nevertheless, there are plentiful opportunities for improvements. For example, the main downside of our method is its performance in relation to time, which may be enhanced by introducing social rules, such as a preferred side to avoid different teams; properly balancing the shared avoidance effort among teams, in a similar fashion to the *Optimal Reciprocal Collision Avoidance* method [van den Berg et al., 2011]; or by relying on different approaches for achieving cohesion and biasing velocities (e.g., [Kimmel et al., 2012; He and van den Berg, 2013]). Investigations along these lines may lead to interesting results that could further extend the velocity obstacle framework.

Because segregation is not a complete task in itself, we see our solutions as tools that can be used as subtasks in complex missions. Examples of which include, but are not limited to, traffic control protocols in structured environments, formation control mechanisms for terrestrial and aerial robots, and splitting behaviors for teams of agents that reach branching paths in an exploration scenario. Furthermore, it is clear that both of our solutions require improvements in order to be suitable for real swarm systems. However, they can be directly used without much modifications in an animation system, as an artificial intelligence behavior for virtual agents, and even on real robots working in an environment that is small when compared to the capabilities of their sensors.

With the results of our work, we intend to increase the interest in research on heterogeneous robotic swarms, and our goal in this direction is to explore distinct behaviors that are exclusive to such systems. Heterogeneity is a fundamental characteristic in many natural systems, and its use on robotic swarms may lead to the design of better, robust, and reliable systems.

Bibliography

Abe, Y. and Yoshiki, M. (2001). 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.

Agarwal, P. (1995). Cellular segregation and engulfment simulations using the cell programming language. *Journal of Theoretical Biology*, 176(1):79–89.

Alonso-Mora, J., Breitenmoser, A., Beardsley, P., and Siegwart, R. (2012). Reciprocal collision avoidance for multiple car-like robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 360 –366.

Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., and Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple non-holonomic robots. In *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 203–216. Springer Berlin Heidelberg.

Ame, J.-M., Rivault, C., and Deneubourg, J.-L. (2004). Cockroach aggregation based on strain odour recognition. *Animal Behaviour*, 68:793–801.

Balch, T. and Arkin, R. C. (1998). Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939.

Balch, T. and Hybinette, M. (2000). Social potentials for scalable multirobot formations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 73–80.

Barca, J. C. and Sekercioglu, Y. A. (2013). Swarm robotics reviewed. *Robotica*, 31(3):345–359.

Barnes, L., Fields, M., and Valavanis, K. (2009). Swarm formation control utilizing elliptical surfaces and limiting functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1434–1445.

Belta, C. and Kumar, V. (2004). Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875.

Bennewitz, M., Burgard, W., and Thrun, S. (2002). Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, USA.

Borenstein, J. and Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 7(3):278 –288.

Brooks, R. A. (1983). Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2):190–197.

Burton, T., Kane, B., Waters, D., and Hamm, S. (1992). Batman returns. Warner Bros. Film (126 min).

Cao, Y., Fukunaga, A., Kahng, A., and Meng, F. (1995). Cooperative mobile robotics: antecedents and directions. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 226–234.

Chaimowicz, L. and Kumar, V. (2007). Aerial shepherds: Coordination among UAVs and swarms of robots. In *Distributed Autonomous Robotic Systems 6*, pages 243–252. Springer Japan.

Chaimowicz, L., Michael, N., and Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2498–2503.

Chen, J., Gauci, M., Price, M. J., and Groß, R. (2012). Segregation in swarms of e-puck robots based on the brazil nut effect. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 163–170.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, USA.

Connolly, C. I. (1992). Applications of harmonic functions to robotics. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 498–502.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3 edition.

Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. In *From animals to animats: Proceedings of the 1st International Conference on Simulation of Adaptive Behavior*, pages 356–363, Cambridge, MA, USA. MIT Press.

Desaraju, V. R. and How, J. P. (2011). Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In *Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4956–4961.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Dorigo, M., Floreano, D., Gambardella, L., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., Brambilla, M., Brutschy, A., Burnier, D., Campo, A., Christensen, A., Decugniere, A., Di Caro, G., Ducatelle, F., Ferrante, E., Forster, A., Martinez Gonzales, J., Guzzi, J., Longchamp, V., Magnenat, S., Mathews, N., Montes de Oca, M., O'Grady, R., Pinciroli, C., Pini, G., Retornaz, P., Roberts, J., Sperati, V., Stirling, T., Stranieri, A., Stutzle, T., Trianni, V., Tuci, E., Turgut, A., and Vaussard, F. (2013). Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Magazine*, 20(4):60–71.

Dresner, K. and Stone, P. (2005). Multiagent traffic management: An improved intersection control mechanism. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 471–477.

Eberhart, R. C., Shi, Y., and Kennedy, J. (2001). *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559.

Eduard, B. and Wilkinson, D. G. (2012). Molecular mechanisms of cell segregation and boundary formation in development and tumorigenesis. *Cold Spring Harbor Perspectives in Biology*, 4(1):1–14.

Egerstedt, M. and Hu, X. (2001). Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951.

Erdmann, M. and Lozano-Perez, T. (1986). On multiple moving objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1419–1424.

Ferner, C., Wagner, G., and Choset, H. (2013). ODrM* optimal multirobot path planning in low dimensional search spaces. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3854–3859.

Fiorini, P. and Shillert, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17:760–772.

Foty, R. A. and Steinberg, M. S. (2005). The differential adhesion hypothesis: a direct evaluation. *Developmental Biology*, 278:255–263.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33.

Fraichard, T. and Asama, H. (2003). Inevitable collision states. a step towards safer robots? In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 388–393.

Franks, N. and Sendova-Franks, A. (1992). Brood sorting by ants: distributing the workload over the work-surface. *Behavioral Ecology and Sociobiology*, 30(2):109–123.

Fulgenzi, C., Spalanzani, A., and Laugier, C. (2007). Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1610–1616.

Garcia, R., Shiroma, P., Chaimowicz, L., and Campos, M. (2007). Um arcabouço para a localização de enxames de robôs. In *Anais do VIII Simpósio Brasileiro de Automação Inteligente.*

Gerkey, B. P., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11^{th} International Conference on Advanced Robotics*, pages 317–323.

Groß, R., Magnenat, S., and Mondada, F. (2009). 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.

Grossman, D. (1988). Traffic control of multiple robot vehicles. *IEEE Journal of Robotics and Automation*, 4(5):491–497.

Guy, S. J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., and Dubey, P. (2009). Clearpath: highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

He, L. and van den Berg, J. (2013). Meso-scale planning for multi-agent navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2839–2844.

Hennes, D., Claes, D., Meeussen, W., and Tuyls, K. (2012). Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 147–154.

Holden, N. and Freitas, A. A. (2004). Web page classification with an ant colony algorithm. In *Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 1092–1102. Springer Berlin Heidelberg.

Hou, S., Cheah, C., and Slotine, J. (2009). 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.

Howard, A., Matarić, M. J., and Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Syst. 5*, pages 299–308. Springer Japan.

Jackson, P., Tolkien, J. R. R., Walsh, F., and Boyens, P. (2001). The lord of the rings: The fellowship of the ring. New Line Cinema. Film (176 min).

Kamphuis, A. and Overmars, M. H. (2004a). Finding paths for coherent groups using clearance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 19–28.

Kamphuis, A. and Overmars, M. H. (2004b). Motion planning for coherent groups of entities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3815–3822.

Kant, K. and Zucker, S. W. (1986). Toward efficient trajectory planning: the path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89.

Karaman and Frazzoli (2011). Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894.

Kato, S., Nishiyama, S., and Takeno, J. (1992). Coordinating mobile robots by applying traffic rules. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1535–1541.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 12(4):566–580.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948.

Khatib, O. (1985). 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.

Kimmel, A., Dobson, A., and Bekris, K. (2012). 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.

Kloder, S. and Hutchinson, S. (2006). Path planning for permutation-invariant multi-robot formations. *IEEE Transactions on Robotics*, 22(4):650–665.

Koren, Y. and Borenstein, J. (1991). 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.

Kumar, M., Garg, D., and Kumar, V. (2010). Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3):743–748.

Latombe, J.-C. (1991). *Robot Motion Planning.* Kluwer Academic Publishers, Norwell, MA, USA.

LaValle, S. M. (2006). *Planning Algorithms.* Cambridge Univ. Press, Cambridge, U.K.

LaValle, S. M. and Hutchinson, S. A. (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912–925.

LaValle, S. M. and Kuffner Jr., J. J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400.

Leonard, N. E. and Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2968–2973.

Li, T.-Y. and Chou, H.-C. (2003). Motion planning for a crowd of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 4215–4221.

Lien, J.-M., Bayazit, O., Sowell, R., Rodriguez, S., and Amato, N. (2004). Shepherding behaviors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4159–4164.

Lozano-Perez, T. (1983). Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120.

Luca, A. D., Oriolo, G., and Vendittelli, M. (2000). Stabilization of the unicycle via dynamic feedback linearization. In *6th IFAC Symp. on Robot Control*, pages 397–402.

Lyapunov, A. M. (1992). The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534.

Marcolino, L. and Chaimowicz, L. (2009). 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.

Martel, S. and Mohammadi, M. (2010). Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 500–505.

Michael, N., Belta, C., and Kumar, V. (2006). Controlling three dimensional swarms of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 964–969.

Michael, N. and Kumar, V. (2009). Planning and control of ensembles of robots with non-holonomic constraints. *Int. Journal of Robotics Research*, 28(8):962–975.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). 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.

Narain, R., Golas, A., Curtis, S., and Lin, M. C. (2009). Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics*, 28(5):122:1–122:8.

O'Donnell, P. and Lozano-Perez, T. (1989). Deadlock-free and collision-free coordination of two robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 484–489.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420.

Parker, L. E. (2008). Distributed intelligence: overview of the field and its application in multi-robot systems. In *Journal of Physical Agents*, volume 2, pages 5–14.

Pimenta, L., Pereira, G., Michael, N., Mesquita, R., Bosque, M., Chaimowicz, L., and Kumar, V. (2013). Swarm coordination based on smoothed particle hydrodynamics technique. *IEEE Transactions on Robotics*, 29(2):383–399.

Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., and Campos, M. F. M. (2005). On computing complex navigation functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3452–3457.

Pimenta, L. C. A., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. (2008). Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3947–3952.

Reif, J. H. (1979). Complexity of the mover's problem and generalizations. In *Proceedings of the 20th Annual Symp. on Foundations of Computer Science*, pages 421–427.

Reif, J. H. and Wang, H. (1999). Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27(3):171–194.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34.

Ridley, M. (2003). *Evolution.* Wiley-Blackwell, 3rd edition.

Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.

Saha, M. and Isto, P. (2006). Multi-robot motion planning by incremental coordination. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5960–5963.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin Heidelberg.

Santos, V., Campos, M., and Chaimowicz, L. (2012). On segregative behaviors using flocking and velocity obstacles. In *Proceedings of the 11th International Symposium on Distributed Autonomous Robotic Systems*, pages 377–389.

Santos, V. and Chaimowicz, L. (2011a). Hierarchical congestion control for robotic swarms. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4372–4377.

Santos, V. and Chaimowicz, L. (2011b). Uso de hierarquias no controle de enxames robóticos. In *Anais do X Simp. Brasileiro de Automação Inteligente*, pages 557–562.

Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA.

Snape, J., van den Berg, J., Guy, S., and Manocha, D. (2011). The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27(4):696–706.

Steinberg, M. S. (1963). Reconstruction of tissues by dissociated cells. *Science*, 141:401–408.

Stentz, A. (1993). Optimal and efficient path planning for unknown and dynamic environments. Technical report CMU-RI-TR-93-20, Robotics Institute, Carnegie Mellon University.

Tan, K.-H. and Lewis, M. (1996). 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.

Tanner, H., Pappas, G., and Kumar, V. (2004). Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455.

Tanner, H. G., Jadbabaie, A., and Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868.

Thalmann, D. and Musse, S. R. (2007). *Crowd Simulation*. Springer-Verlag, London, United Kingdom, 1 edition.

Turpin, M., Michael, N., and Kumar, V. . (2013a). Trajectory planning and assignment in multirobot systems. In *Algorithmic Foundations of Robotics X*, volume 86 of *Springer Tracts in Advanced Robotics*, pages 175–190. Springer Berlin Heidelberg.

Turpin, M., Michael, N., and Kumar, V. . (2013b). Concurrent assignment and planning of trajectories for large teams of interchangeable robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 834–840.

van den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 3–19. Springer Berlin Heidelberg.

van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1928–1935.

van den Berg, J. and Overmars, M. (2005). Prioritized motion planning for multiple robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 430–435.

van den Berg, J., Snape, J., Guy, S. J., and Manocha, D. (2012). Reciprocal collision avoidance with acceleration-velocity obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3475–3482.

Viswanath, D. V. K. and Krishna, K. M. (2007). Sensor network mediated multi robotic traffic control in indoor environments. In *Proceedings of the 13^{th} International Conference on Advanced Robotics*, pages 1220–1225.

Wagner, G. and Choset, H. (2011). M*: A complete multirobot path planning algorithm with performance bounds. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3260–3267.

Wagner, G., Kang, M., and Choset, H. (2012). Probabilistic path planning for multiple robots with subdimensional expansion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2886–2892.

Warren, C. (1990). Multiple robot path coordination using artificial potential fields. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 500–505.

Wilkie, D., van den Berg, J., and Manocha, D. (2009). Generalized velocity obstacles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5573 –5578.