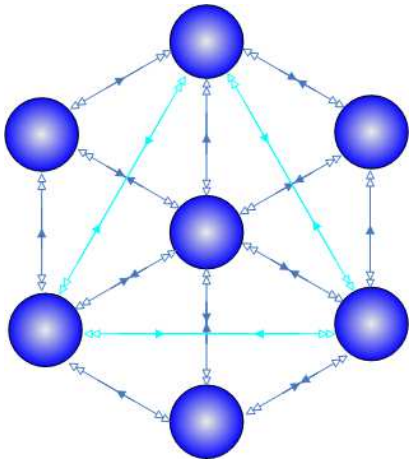
Universidade Técnica de Lisboa

Instituto Superior Técnico



Non-Holonomic Robot Formations with Obstacle Compliant Geometry

Pedro Viçoso Fazenda

(Licenciado)

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

DOCUMENTO PROVISÓRIO

Lisboa, Janeiro de 2008

Resumo

A presente dissertação apresenta uma metodologia para controlar uma formação de veículos holonomicos e/ou não-holonomicos, com uma geometria deformável e adaptável aos obstáculos vizinhos (onde se inclui também, para cada robô, os restantes membros da formação). Os veículos estão virtualmente interligados por influência de campos de potencial artificiais, que estabilizam assimptoticamente a formação e que mantêm os vários robôs da formação separados por distâncias especificadas. Um veículo líder seleccionado da equipa, ou um ponto de referência, será usado para conduzir o resto da equipa através de uma área repleta de obstáculos até um determinado ponto destino. Cada veículo tem acesso às posições de todos os restantes elementos da sua equipa, e capacidade de detectar os obstáculos que existem na sua área envolvente. Todos os robôs irão tentar manter a distância especificada até ao líder e aos restantes membros da formação, estando sujeitos a forças de atracção e repulsão. As forças derivam dos gradientes negativos dos campos de potencial que os interligam com a formação. O procedimento garante a coesão da formação sem colisões entre os membros participantes. Para evitar colisões com obstáculos, os veículos estão sujeitos adicionalmente a campos de potencial de repulsão que os afastam daqueles, mas que levantam problemas devido à criação de mínimos locais. Para evitar que fiquem bloqueados nestes pontos cada veículo irá recordar as últimas n posições do líder e usar esta informação para contornar os obstáculos e manter a formação.

Palavras Chave: Controlo de Formações, Campos de Potencial, Evitar Obstáculos, Robótica Distribuída, Robótica Cooperativa, Sistemas Multi-Robot.

Abstract

This thesis introduces a method to control a formation of holonomic or non-holonomic vehicles with a deformable geometry, compliant with nearby obstacles (including those represented by each robot team-mates). The vehicles are virtually linked to each other by the influence of artificial potentials that asymptotically stabilize the formation and keep all the robots separated by specified distances. A leader selected from the team, or a virtual reference point, is used to guide the team of autonomous vehicles throughout an area scattered with obstacles. Each vehicle has access to the positions of all its team-mates, and senses the obstacles within a limited range of its neighbourhood. All robots attempt to maintain the specified distances to the leader and every other member of the formation, as the result of attractive and repulsive forces. The forces are the negative gradients of the potential fields that interconnect the formation vehicles. The procedure guarantees the cohesion of the formation without collisions between the participating members. To avoid collisions with obstacles, the vehicles are subjected to extra repulsive potentials, which induce problems due to local minima. To avoid getting stuck in those points, each vehicle recalls the n latest positions of the leader and uses this information to move around the obstacle and stay in formation.

Keywords: Formation Control, Potential Fields, Obstacle Avoidance, Distributed Robotics, Cooperative Robotics, Multi-Robot Systems.

Preface and Acknowledgements

I am indebted to many colleagues and friends who supported and encouraged me making the realization of this work possible. In particular, to my thesis supervisor Professor Pedro Lima.

I would like to express my gratitude to all my colleagues at ISEL/CEDET (*Instituto Superior de Engenharia de Lisboa/Centro de Estudos e Desenvolvimento de Electrónica e de Telecomunicações*), specially to António Couto Pinto, Helena Sousa Ramos, Miguel Campilho Gomes, João Casaleiro, Tiago Oliveira, Ricardo Reis, Carlos Carvalho, José Rocha, José Nascimento, Eugénio Furtado, Vitor Costa and Rui Duarte.

I would also like to thank ISR/ISLAB (Institute for Systems and Robotics/ Intelligent Systems Lab) and all my friends at this lab, for providing me with excellent working conditions and a very productive and fertile environment.

Finally, I would like to thank my family and my wife and once again, to my supervisor Pedro Lima, who read through the manuscript before I submitted it.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms and Abbreviations

SI	Swarm Intelligence
FF	Formation Framework
VL	Virtual Leader
HP	Holonomic Point
PFM	Potential Fields Methods
APF(s)	Artificial Potential Field (s)
GVF	Gradient Vector Field
IF(s)	Interaction Force (s)
IP(s)	Interaction Potential(s)
ODL	Obstacle Detection Layer
CAP	Current Attraction Point
NF(s)	Navigation Function(s)
w.r.t.	with respect to
i.e.	from the Latin *id est* meaning "that is"
e.g.	from the Latin *exempli gratia* meaning "for example"
etc	from the Latin *et cetera* meaning "and other things"

List of Symbols

E^n	Euclidean n-space
W	Bounded 2D or 3D world
SW	World with Spherical Boundaries in E^n
STW	Star World
$\mathfrak{F}$	Free Configuration Space
$\partial\mathfrak{X}$	Boundary of a set, $\mathfrak{X} \subseteq E^n$
$\mathbf{F}$	Force Vector
U	Artificial Potential Field
V	Vehicle-to-Vehicle Interaction Potential
U_{att}	Attractive Potential
U_{rep}	Repulsive Potential
C	Configuration Space
$\mathbf{q}$	Configuration Vector that captures the state of a robot in W
q	Magnitude of vector $\mathbf{q}$, $q = \|\mathbf{q}\|_2$ (Euclidean norm)
$\hat{\mathbf{q}}$	Normalized vector, $\hat{\mathbf{q}} = \frac{\mathbf{q}}{q}$ for $q \neq 0$
$\mathbf{f_{vi}}$	Control dissipative force
κ	Magnitude of $\mathbf{f_{vi}}$
$A(\mathbf{q_i})$	Points in W that are occupied by the robot i when it is in configuration $\mathbf{q_i}$
O	Obstacle region in the world

$\mathbf{v_i}$	Linear velocity vector of robot i
$\mathbf{u_i}$	Control force applied to robot i
θ_i	Orientation of robot i

1 Introduction

*Given some task specified by a designer, a multiple-robot system displays cooperative behavior if, due to some underlying mechanism (*i.e.*, the "mechanism of cooperation"), there is an increase in the total utility of the system...*[1]

1.1 Motivation

The introduction section of many articles on multiple robot formation control presents the same recurrent arguments about the advantages of using multiple cooperating robots. This is an appealing topic in the robotic community because some tasks can be better accomplished in space and time when several robots cooperate, than when using a single monolithic robot. This is particularly relevant when the mission is to be accomplished over a remote and extensive area, where some tasks cannot be carried out by a single robot. A team of robots can expand the spatial coverage of the service area, with the increase of service capacity and margins of success, due to the redundancy of having several nodes performing the same task, as opposed to a single robot mission [2].

The concepts explored in this thesis are borrowed from nature, following examples like ant colonies, bird flocking, animal herding, bacteria molding and fish schooling,

and are usually grouped in the area of Swarm Intelligence (SI). SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. Although there is normally no centralized control structure dictating how individual agents should behave, local interactions among such agents often lead to the emergence of global behavior. The simple rules by which individuals interact can generate complex group behavior. Nature favors this collective interaction. Animals that can combine sensing abilities can do better in avoiding predators and foraging for food. Studies have shown that geese flying in "V" or wedge formations can extend their range due to the energy savings from riding each other's air vortexes. Dolphins are known to swim in formations to protect their calves from predators. Herd and pack animals have been shown to use group organization to attack large prey, defend against predators, defend their territory and increase their chances of survival.

Robots are natural candidates to demonstrate the concepts studied in SI systems. Each robot will follow a set of rules influenced by locally available information, and emergent behavior can be used to help managing some mission tasks. The idea is not necessarily to get rid of the human operators, but to have a human-in-the-loop with some control over the collective behavior, managing a specified mission from a higher and simplified level. A centralized control center can this way delegate the processing of vast amounts of information to local agents and treat the entire formation as a unique body. This abstraction starts at the lowest level of the multi-robot coordination strategy that includes path planning and inter vehicle collision avoidance. Each robot reacts locally to the environment and guaranties the necessary inter-vehicle distance to near neighbors. Higher level commands to the formation can include, for example, commands to expand, contract, or move to a specified area. The topic is promising for a wide variety of applications ranging from team search and rescue operations, space exploration, feature tracking and sampling, mine detection, fire monitoring and tracking with aerial vehicles, to meteorological or oceanographic surveys. In general, any application where the spatial and temporal scales presented in the field of interest are so dynamic, that a single vehicle cannot provide adequate sampling [3]. Having a human operator managing each robot individually, at the lowest level, in many of these situations, can become quite restrictive and expensive because this would imply the need for some requirements like, *e.g.*, having several human operators operate the formation, and a real-time communication link between each robot and the control station, with a certain latency limitation requirement.

To really explore the full advantages of swarms, the control approach should scale with the number of robots. The work presented in this thesis focuses on developing a control framework for coordinating groups of autonomous robots, where a control law for each robot will respond to the robot's most *basic behaviors*, *i.e.*, maintain group cohesion and avoid inter-robot collisions and collisions with obstacles. The expected outcome should be a *Formation Framework* (FF) structure that can be used to support many of the applications stated above.

1.2 Problem Overview

When working with *Multi-Robot Systems* an important mechanism to generate cooperation, namely, task decomposition and allocation, can be simplified if the applications involve space coverage (in 2D or 3D space). If each robot is aimed to analyze a certain portion of space, the FF may contribute to this mechanism in the sense that it will distribute and provide means to regulate the distribution of such agents throughout the area of interest.

In general, all of the work in cooperative mobile robotics began after the introduction of the new robotics paradigm of behavior-based control, rooted in biological inspirations [4]. In various biological societies it seems that simple local control rules include, for example, keeping a certain distance to nearest neighbors and maintaining group cohesion. This behavior can be extended for purposes such as distributing and arranging teams of robots over some selected working area. The work presented in this thesis concerns a team of robots, with an initial configuration, which must be deployed to a certain destination zone. To accomplish this task a robot, selected as a leader, or a virtual reference point, is used to guide the whole team towards the desired destination, by tracking a previously planned path. The surrounding field can be scattered with obstacles, and each robot must try to keep in formation avoiding collisions between these obstacles and its team-mates.

When embedding robot tasks in a two or three dimensional world the issue will include multi-agent path planning, moving to formation and pattern generation. The idea is to delegate these issues down in the control hierarchy to become basic and controllable behaviors, namely, the ones stated above. Path planning, moving to formation and pattern generation, would arise naturally from the need that each agent has to maintain group cohesion and avoid collisions. The formation patterns will result from how these behaviors are implemented and, for the purpose of this work, lack

the need for any kind of specific formation arrangement. We will focus on "natural" models of group motion, as opposed to more structured models of coordinated motion (such as moving with some arbitrary random geometry).

It is assumed that vehicles have knowledge of the positions of their teammates and sense obstacles within a limited range of their neighborhood. They behave in accordance with their surrounding environment, by considering the sensed obstacles, and with the position of their nearest neighbors. For a static leader the formation should converge asymptotically to a stable configuration, with the vehicles resting at a certain distance from each other. By moving the leader the rest of the formation will follow, causing a mass movement of the swarm, acting upon the need of regaining this equilibrium configuration. This global dependence on the virtual leaders motion reduces the task planning problem of multiple collision free paths for many vehicles, to planning just one collision free path - the leader's path. Figure 1.1 shows a representation of a team, with the leader represented in red, being deployed to a target area.

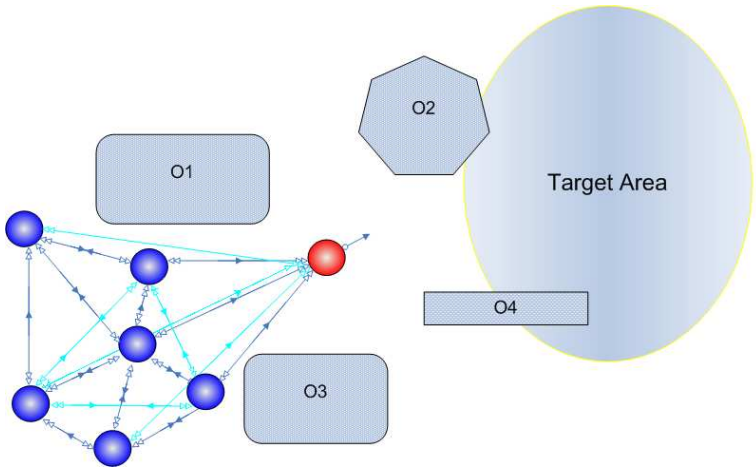


Figure 1.1: Deploying a formation throughout a field of obstacles($O_1, \ldots, O_4$), towards a target area. Arcs represent force links connecting the robots.

A remote operator can also guide the team by sending control instructions to the leader (or by controlling the virtual reference point). Regardless of how the leader moves, the team must keep up to maintain group cohesion. This may not be a trivial task because in some situations a part, or even the whole formation, may get stuck behind obstacles. Considering the point of view and interest of a remote operator,

the desired behavior of the formation would be for the operator to just trace a path (or drive the leader to the destination area), knowing that the rest of the formation will manage to follow unattended. The idea for this thesis is centered on building a framework that is inspired by the idea of using simple local control rules for the proposed tasks. The complexity of the applications may then be extended, by building upon this framework.

1.3 Related Work

1.3.1 Modeling the Formation

The work on *swarm* systems began as a work on Cellular Robotic Systems, where many simple agents occupied one or two-dimensional environments and were able to perform tasks such as pattern generation and self-organization [5]. Prior to the 80's, much of the research in the field of robotics was concentrated on single robot systems. Since then, the field of distributed robotics with multiple mobile robots has grown and, with it, the study of motion coordination which includes multi-robot path planning, traffic control, formation generation and formation keeping. The examples of cohesive group movement and coordination, that can be found throughout the natural world, have always raised much interest in the scientific community. Mathematical biologists have attempted to model this swarming behavior for some time and, in many cases, the behavior has been reduced to rules of attraction and repulsion between neighbors see, *e.g.*, [6],[7] and [8]. The potential field methods pioneered by Kathib [9], developed from an analogy with electrostatic field physics, were used for building such social rules. Many articles have been written on this subject since then. Examples follow. Gazi and Passino [10], introduced a model and a study of its stability for individuals considered as points that move in space, according to an attractant/repellent or nutrient profile *i.e.*, attraction to other individuals on long distances, repulsion from the others at short distances, and attraction/repulsion to more favorable/unfavorable regions. Several profiles were modeled and studied, but none of them model obstacles. The most complicated forms that were presented are multimodal gaussian profiles, that may have multiple extremum points. These profiles may break group cohesion, and global convergence of the formation to a certain attraction point is not guaranteed. No solution was presented on how to solve this limitation. Zavlanos and Pappas [11] considered the problem of controlling a network of agents,

so that the resulting motion always preserves the connectivity of the network, that is described by a dynamic graph. *Artificial Potential Fields* (APFs) are used to drive the agents to configurations away from the undesired space of disconnected networks, while avoiding collisions with each other. The solution does not mention how connectivity is preserved in the presence of obstacles. Tanner *et al.* [12],[13], in an article on stable flocking of mobile agents, present a formation using inter-vehicle potentials and the study on its stability. A set of control laws is presented that give rise to flocking behavior an provide a system theoretic justification, by combining results from classical control theory, mechanics and algebraical graph theory. Navigation in an obstacle field was not considered, nor the impact that any obstacle may have on the flocking behavior. Yamagishi [14] presented more work on controllers that utilize nearest-neighbor artificial potentials for collision-free formation maintenance. Each agent senses the environment and performs gradient climbing tracking of, *e.g.*, a hazardous chemical spilled over a certain area. Robots are considered points in a space without obstacles. Howard *et al.* [15], showed a different approach on how to use APFs for sensor network deployment. Several nodes are self-deployed, starting from an initial compact configuration, and spread out to cover a certain area. Each node is repelled by both obstacles and by other nodes. Group cohesion is not maintained and nodes tend to expand over the working area. No mention was made on how the nodes can regroup after being deployed and, by the examples given, many nodes will probably get trapped behind walls and other obstacles.

This thesis follows some of the ideas presented by Ögren, Fiorelli and Leonard [16], [3],[17],[18], in particular, the equation that defines the APF that virtually links vehicles to each other in the formation. Their work is extended to handle non-holonomic vehicles, avoid obstacles, and to provide a systematic method of avoiding local minima of the *Interaction Potentials* (IPs), which would get some of the formation robots stuck in obstacles of particular geometric configurations. Elkaim *et al.* [19],[20] and [21], presented a multi-vehicle control framework very similar to the one presented in this thesis, but do not present a clear and usable solution to apply the results to non-holonomic vehicles. Instead, to reach an equilibrium point located sideways to the former vehicles, on a area not reachable without maneuvering, due to kinematic constraints, robots use inverse sinusoid trajectories. To reach these points, vehicles have to drive forward and backwards towards the desired position. For obstacle avoidance, a global knowledge of the world map is assumed where obstacles are enclosed by bounding convex polygons. The leader must move in such a manner that enables

the entire formation to keep-up and, no reference is made to the underlying problem of local minima.

1.3.2 Formation Navigation

Khatib originally developed APFs as an online collision-avoidance approach and many articles have been written on this subject, see, *e.g.*, [22],[23],[24]. The key limitation to this approach is that robots get often trapped in a local minimum either than the desired goal configuration. Koren and Borenstein [25] identified and criticized inherent limitations of the *Potential Field Methods* (PFM) for mobile robot navigation. Some work has been done on formation navigation and obstacle avoidance see, *e.g.*, [26] and [27]. Many articles assume a prior global knowledge of the world map, or do not assume local minima in the context of the objectives that are presented in this thesis. Tanner [28] introduces a set of nonsmooth control laws that enable a group of vehicles to synchronize their velocity vectors and move as a flock while avoiding collisions with each other and with static obstacles in their environment. The problem of escaping singularities was not fully covered. Ögren and Leonard [29] presents an approach to obstacle avoidance for a group of vehicles moving in formation through a partially unknown environment. In this article the leader will have to attend to the rest of the formation before taking any actions, to be confident that none of the followers will collide with an obstacle. The presented solution will most likely have problems scaling to larger formations.

From some of the studied techniques considered for local minimum avoidance is the extensive work of Kodischek and Rimon [30]. Using geometrical arguments they showed that, at least in certain types of domains, a special class of potential functions called navigation functions could be built that are exempt of local minima either than the arrival configuration. However, their extension to more general spaces proved to be quite difficult. The ideas were initially well received for this thesis because they represent the very best that PFM has to offer, without the local minimum limitation. It was then verified that, although the presented equations are theoretically interesting, they present limitation for practical implementation. They are hard to build on general spaces and, even in the domains that the authors use as models, the potential functions tend to become flat near and far away from the goal, with sharp transitions in between. This presents limitations when implementing the gradient descent approach, due to numerical errors associated with the gradient values.

1.4 Goals and Contributions

The main goals and contributions of this work are:

- Describe and analyze a Formation Control Framework that can be used systematically in a procedure for deploying and working with robot formations.
- Extend the ideas of the work of N.Leonard and co-worker to non-holonomic vehicles.
- Introduce methods to avoid local minima problems within the formation in the presence of non-convex obstacles.
- Build a simulation environment to test the algorithms. Simulate several scenarios and formations with the environment and discuss the results.

1.5 Thesis Outline

The remainder of this document is organized as follows:

Chapter 2, Formation Framework - describes the Formation Framework using different artificial potentials and holonomic vehicles. Then, a suitable transformation is used to control a holonomic point on the vehicle, so that the formation control framework, originally developed for vehicles subjected to holonomic restrictions only (also known as holonomic vehicles), can be extended to non-holonomic vehicles. In the end, a scalable solution for larger formations is presented.

Chapter 3, Obstacle Avoidance - describes the full formation control algorithm, endowing the original formulation with obstacle avoidance that handles local minima in the interaction potentials. It ends with a discussion of some aspects that can be explored in future work.

Chapter 4, Simulation Setup - a simulation setup using the Matlab environment is presented, followed by the description of a set of tests, aimed at validating the theoretical ideas presented in the former chapters.

Chapter 5, Results - presents the results of each one of the tests described in Chapter 4, followed by their evaluation.

Chapter 6, Conclusions and Future Work - the main conclusions of this work are drawn, as well as remarks for future work. This includes other formation structures that can be explored, and some ideas on the possible application of Navigation Functions in the formation for obstacle avoidance and navigation.

2

Formation Framework

Self-organization in a SWARM is the ability to distribute itself "optimally" for a given task, e.g.*, via geometric pattern formation or structural organization...*[1]

Following the example of what happens in nature with animals in a flock or herd, at the most basic control level, vehicles can arrange themselves into a dynamic formation to move throughout space. Each team member must constantly consider all nearby teammates and maintain a particular set of geometric constraints among themselves. This chapter discusses the multi-vehicle control framework used to coordinate the vehicles into regular formations, using APFs. An APF is defined for each pair of interacting members of the formation, generating a *virtual force* that is related to their inter-vehicle distance. The vehicles will act to follow a steepest descent path towards the geometric point at which the sum of the overall actuating forces becomes zero. The main motivation for this approach derives from seeking closed form mathematical expressions to encode actuator commands, rather than algorithms which include logical decisions. Moreover, following the guideline of having simple rules by which individuals interact, and considering that the most basic behavior in a formation is directed towards maintaining group cohesion and inter-vehicle spacing, the solution

seems to fit quite adequately the specifications described in the previous chapter.

Throughout the chapter we will describe the APF methods and the FF. We will show how to move the formation and how to create heterogeneous formations, using vehicles with different locomotion schemes, by extending the ideas for non-holonomic differential-drive vehicles. Then, we will describe how the framework scales up well and, in the end, present and discuss other formation structures for future work.

2.1 Potential Fields Methods

Assume that a robot operates in a bounded 2D or 3D world, $W \in E^N$, such that $N = 2$ or $N = 3$. An n-dimensional *configuration vector*, $\mathbf{q_i} = (x_1, \ldots, x_n)$, captures the position, orientation, joint angles, and other information related to the state of the robot i in W. Let $\mathbf{q_{ij}} = \mathbf{q_i} - \mathbf{q_j} \in \mathbb{R}^n$. Throughout this thesis, **bold** characters indicate column vectors and the same variable in italic font refers to its magnitude, for example $q_{ij} = \|\mathbf{q_{ij}}\|_2$. The hat character indicates that a vector has been normalized, *e.g.*

$$\hat{\mathbf{q}}_{\mathbf{ij}} = \frac{\mathbf{q_{ij}}}{q_{ij}} \text{ for } q_{ji} \neq 0$$

Denote the boundary of a set, $\mathfrak{X} \subseteq E^n$, by $\partial\mathfrak{X}$. Let C be the *configuration space* (*i.e.*, the set of all possible configurations). Let $A(\mathbf{q_i})$ denote the set of points in W that are occupied by the robot i when it is in configuration $\mathbf{q_i}$. Let $O \in W$, denote an *obstacle region* in the world. Let $\mathfrak{F}$ denote the *free configuration space*, *i.e.*, the set of configurations, $\mathbf{q_i}$, such that $A(\mathbf{q_i}) \cap O = 0$. In the PFM a robot, with a certain configuration $\mathbf{q_i} \in \mathfrak{F}$, is treated like a particle under the influence of an APF U. The idea of using potential functions for the specification of robot tasks was pioneered by Khatib [9] in the context of obstacle avoidance. They present the utility of automatically translating a robot tasks description into a feedback control law to drive the robot actuators. The potential at each configuration will generate a force $\mathbf{F}$ from the *Gradient Vector Field* (GVF), that tends to attract the robot towards a goal configuration $\mathbf{q_d} = (x_{d1}, \ldots, x_{dn}) \in \mathfrak{F}$, while repelling it from all obstacles in O. The resulting force is given by

$$\mathbf{F} = -\nabla U(\mathbf{q_i}) \tag{2.1}$$

2.1.1 Potential Function

The potential is a smooth real valued function $U : \mathfrak{F} \rightarrow \mathbb{R}$, composed by the combination of the following two potential functions:

- **attractive potential** $U_{att}(\mathbf{q_i})$, associated with $\mathbf{q_d}$
- **repulsive potential** $U_{rep}(\mathbf{q_i})$, associated with the obstacles

The potential is given by

$$U(\mathbf{q_i}) = U_{att}(\mathbf{q_i}) + U_{rep}(\mathbf{q_i}) \tag{2.2}$$

with the following resulting force

$$\mathbf{F}(\mathbf{q_i}) = -\nabla U_{att}(\mathbf{q_i}) - \nabla U_{rep}(\mathbf{q_i}) \tag{2.3}$$

Attractive Potential

In a general sense, the attractive potential $U_{att}(\mathbf{q_i})$ increases as $\mathbf{q_i}$ moves away from $\mathbf{q_d}$. One example, from many possible forms that this function can assume, is the *parabolic well*, given by the following expression:

$$U_{att1}(\mathbf{q_i}) = \frac{1}{2}\xi q_{id}^2 \tag{2.4}$$

where ξ is a positive scaling factor. The function presents a unique minimum at $\mathbf{q_d}$, *i.e.*, $U_{att}(\mathbf{q_d}) = 0$, and increases towards infinite, as the robot moves away from this configuration. From equation (2.1)

$$\begin{aligned} \mathbf{F}_{att1}(\mathbf{q_i}) &= -\nabla U_{att}(\mathbf{q_i}) \\ &= -\nabla \tfrac{1}{2}\xi q_{id}^2 \\ &= -\tfrac{1}{2}\xi(2q_{id})\nabla q_{id} \end{aligned}$$

with $q_{id} = \|\mathbf{q_i} - \mathbf{q_d}\| = (\sum_i (x_i - x_{di})^2)^{\frac{1}{2}}$, we have

$$\begin{aligned} \nabla q_{id} &= \nabla(\textstyle\sum_i (x_i - x_{di})^2)^{\frac{1}{2}} \\ &= \tfrac{1}{2}(\textstyle\sum_i (x_i - x_{di})^2)^{-\frac{1}{2}} \nabla(\sum_i (x_i - x_{di})^2) \\ &= \tfrac{1}{2}(\textstyle\sum_i (x_i - x_{di})^2)^{-\frac{1}{2}} (2(x_1 - x_{d1}), \ldots, 2(x_n - x_{dn})) \\ &= \frac{(x_1,\ldots,x_n)-(x_{d1},\ldots,x_{dn})}{(\sum_i (x_i - x_{di})^2)^{-\frac{1}{2}}} \\ &= \frac{\mathbf{q_i}-\mathbf{q_d}}{\|\mathbf{q_i}-\mathbf{q_d}\|} = \frac{\mathbf{q_{id}}}{q_{id}} = \hat{\mathbf{q}}_{\mathbf{id}} \end{aligned}$$

Thus,

$$\mathbf{F}_{att1}(\mathbf{q_i}) = -\xi q_{id}\hat{\mathbf{q}}_{\mathbf{id}} \tag{2.5}$$

This force is a vector directed towards $\mathbf{q_d}$, with a magnitude linearly related to the distance between both configurations. The force converges linearly to zero, as $\mathbf{q_i}$ approaches $\mathbf{q_d}$, and grows without bound as it moves away.

Another example for the attractive potential function is known as the *conic well*, and is given by the following equation:

$$U_{att2}(\mathbf{q_i}) = \xi q_{id} \tag{2.6}$$

the force vector is given by

$$\mathbf{F}_{att2}(\mathbf{q_i}) = -\xi\frac{\mathbf{q_{id}}}{q_{id}} = -\xi\hat{\mathbf{q}}_{\mathbf{id}} \tag{2.7}$$

and presents a constant magnitude.

To exemplify these potentials consider Figure 2.1, that shows a representation of a 2D world containing the obstacles $O_1, \ldots, O_4 \in O$, and a destination point $\mathbf{q_d}$.

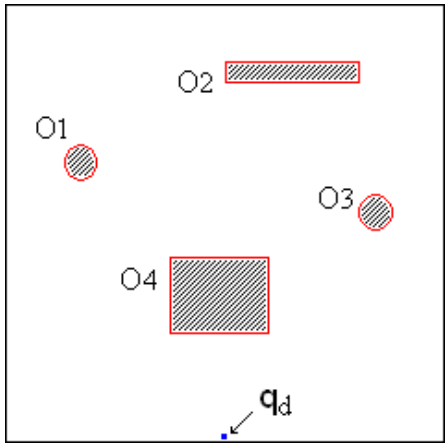


Figure 2.1: 2D World representation example.

The forms of a 3D contour plots of the level curves for both potential functions U_{att1} and U_{att2} are shown in Figures 2.2 and 2.3.

Repulsive Potential

The repulsive potential acts as a penalty function such that any trajectory in W generated by following the negative gradient flow given by equation (2.1), will avoid

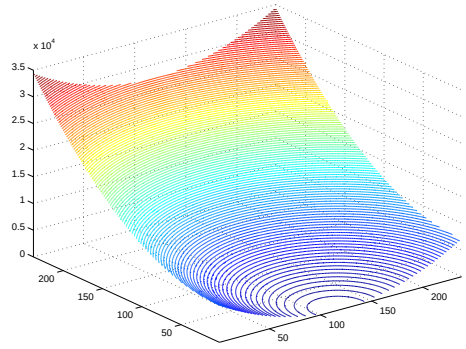


Figure 2.2: Attractive Potential U_{att1}.

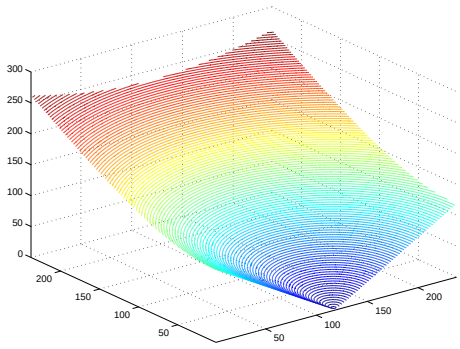


Figure 2.3: Attractive Potential U_{att2}.

all obstacles in O. A repulsive potential U_{rep} is generated, inducing an artificial repulsion from the boundary of the obstacles. As $\mathbf{q_i}$ approaches this boundary, U_{rep} approaches ∞. An example for this potential function is given by the following equation:

$$U_{rep}(\mathbf{q_i}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{q_i})} - \frac{1}{d_l}\right)^2 & \rho(\mathbf{q_i}) \leq d_l \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

where $\rho(\mathbf{q_i})$ is the minimum distance from O to $\mathbf{q_i}$, *i.e.*, $\rho(\mathbf{q_i}) = min_{\mathbf{q_o}\in O}\|\mathbf{q_i} - \mathbf{q_o}\|$, η is a positive scaling factor, and and d_l is the limit distance of the repulsive potential influence. Assuming that O is a single convex region, the repulsive force is given by the following equation:

$$\begin{aligned} \mathbf{F}_{rep}(\mathbf{q_i}) &= -\nabla U_{rep}(\mathbf{q_i}) \\ &= -\nabla(\tfrac{1}{2}\eta(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l})^2) \\ &= -\tfrac{1}{2}\eta\nabla(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l})^2 \\ &= -\eta(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l})\nabla(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l}) \\ &= -\eta(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l})(-1)(\tfrac{1}{\rho^2(\mathbf{q_i})})\nabla\rho(\mathbf{q_i}) \\ &= \eta(\tfrac{1}{\rho(\mathbf{q_i})} - \tfrac{1}{d_l})(\tfrac{1}{\rho^2(\mathbf{q_i})})\nabla\rho(\mathbf{q_i}) \end{aligned} \tag{2.9}$$

With $\mathbf{q_o}$ being the configuration in O closest to $\mathbf{q_i}$, *i.e.*, $\rho(\mathbf{q_i}) = \|\mathbf{q_i} - \mathbf{q_o}\|$, $\nabla\rho(\mathbf{q_i})$ is the unit vector $\hat{\mathbf{q}}_{\mathbf{io}}$ directed away from O along the line passing through $\mathbf{q_o}$ and $\mathbf{q_i}$.

When O is not convex $\rho(\mathbf{q_i})$ becomes a nonsmooth function. Another approach for $U_{rep}(\mathbf{q_i})$ is to consider the contribution of obstacles $O_i \in O$, by associating a repulsive potential to each of them. Each contribution can be weighted by the obstacles size, and the overall repulsive potential will be the sum of all the potentials.

Figure 2.4 shows the truncated form of the 3D contour plot of the level curves for the repulsive potential, associated with the obstacles depicted in the example of

figure 2.1. Figure 2.5 shows the total potential field as defined by equation (2.2), using the parabolic attractive potential.

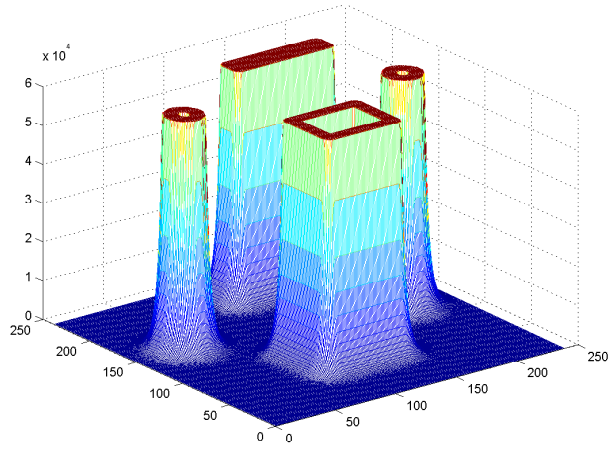


Figure 2.4: Repulsive Potential U_{rep}.

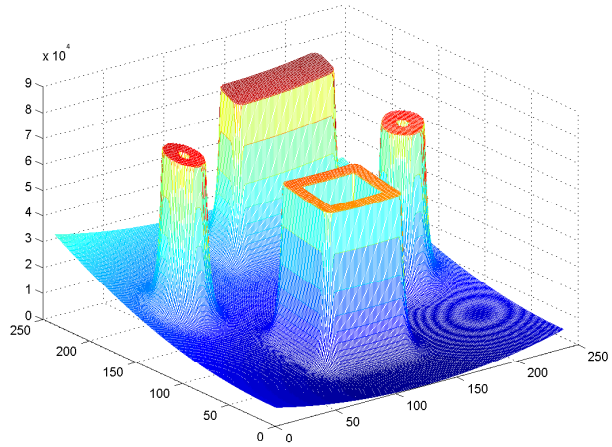


Figure 2.5: Total Potential Field $U = U_{att} + U_{rep}$.

2.2 Team Formation

The formation is composed by a team of N robots. While the presented analysis can be extended to three dimensions, for the purposes of this thesis the problem is limited to a two-dimensional plane. The configuration vector $\mathbf{q_i} \in \mathbb{R}^2$ will represent

the position of the ith vehicle, with $i = 1, \ldots, N$, as shown in figure 2.6. The corresponding velocity vector is given by $\mathbf{v_i} = \dot{\mathbf{q}}_\mathbf{i}$.

The control force on the ith vehicle is given by $\mathbf{u_i} \in \Re^p$. Under full actuation, and assuming unit mass, the vehicle has the following dynamics:

$$\begin{aligned} \dot{\mathbf{q}}_\mathbf{i} &= \mathbf{v_i} \\ \ddot{\mathbf{q}}_\mathbf{i} &= \mathbf{u_i} \end{aligned} \tag{2.10}$$

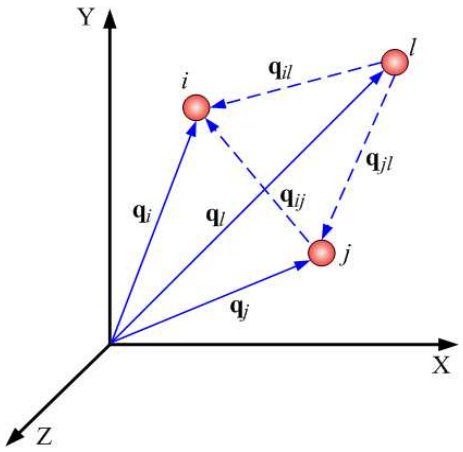


Figure 2.6: The Formation Framework.

It is assumed that the configuration for N vehicles, given by $\mathbf{X} = (\mathbf{q_1}^\mathbf{T}, ..., \mathbf{q_N}^\mathbf{T})$, is known and shared by the whole formation. The control force applied to each vehicle in the formation follows equation (2.1), *i.e.*, the negative gradient of an APF $V_i : \mathbb{R}^{2N} \rightarrow \mathbb{R}^+$ with respect to $\mathbf{q_i}$, plus an extra *control dissipative force* $\mathbf{f_{vi}}$, necessary to provide local asymptotic convergence, given by

$$\mathbf{u_i} = -\nabla_{\mathbf{q_i}} V_i - \mathbf{f_{vi}} \tag{2.11}$$

Force $\mathbf{f_{vi}}$ is made proportional to the vehicle's velocity $\mathbf{v_i}$, *i.e.*, $\mathbf{f_{vi}} = \kappa \mathbf{vi}$.

The potentials couple the dynamics of the vehicles by imposing a desired vehicle-to-vehicle spacing. The overall potential is defined by $V = \sum_{i=1}^{N} V_i$, with $V_i = \sum_{j \neq i}^{N} V_I(\mathbf{q_{ij}}; \mu_I)$. The formation will stabilize in a configuration that minimizes V for a prescribed vector μ of design parameters.

2.2.1 Vehicle-to-Vehicle Interaction

V_I is the the APF that regulates vehicle-to-vehicle interactions. It is a function of the relative distance q_{ij} between the ith and jth vehicle, parameterized by $\mu_I = [\xi_I, d_0, d_1]$. Vehicles interact within a limited range defined by $d_1 > d_0 > 0$. The scalar d_0 specifies the *desired inter-vehicle distance*, with V_I designed to have a global minimum at $q_{ij} = d_0$. The scalar ξ_I is the *multiplicative scaling factor* that adjusts the magnitude of the GVF. Figure 2.7 shows two elements of the formation, i and l, being attracted to their respective equilibrium points represented by $\mathbf{q_d}^i$ and $\mathbf{q_d}^l$, with $\mathbf{q_d}^i = \mathbf{q_i} - \hat{\mathbf{q}}_{\mathbf{ij}}(q_{ij} - d_0)$. Both these point converge to the same unique point.

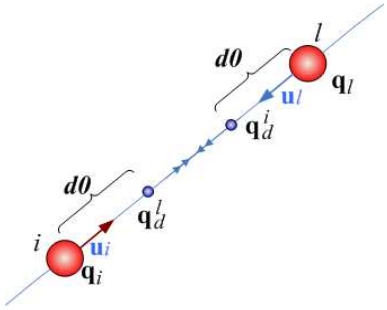


Figure 2.7: Formation attraction to an equilibrium point.

Following equation (2.4), for $0 < q_{ij} < d_1$, a potential that regulates the distance between these two vehicles can assume, from a set of 3 examples that will follow, the following form:

$$\begin{aligned} V_{I1}(\mathbf{q_{ij}}, \xi_I, d_0, d_1) &= \tfrac{1}{2}\xi_I \|\mathbf{q_i} - \mathbf{q_i} + \hat{\mathbf{q}}_{\mathbf{ij}}(q_{ij} - d_0)\|^2 \\ &= \tfrac{1}{2}\xi_I \|q_{ij} - d_0\|^2 \end{aligned}$$

Only local neighbors, within the range defined by d_1, are considered. For the entire configuration space we have

$$V_{I1}(\mathbf{q_{ij}}, \xi_I, d_0, d_1) = \begin{cases} \tfrac{1}{2}\xi_I \|q_{ij} - d_0\|^2 & 0 < q_{ij} \leq d_1 \\ \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.12}$$

The force derived from this potential will be a central force acting along the line connecting the two vehicles. The force regulates relative distance by attracting the vehicles if they are too far apart, or by repelling them, if they are too close together.

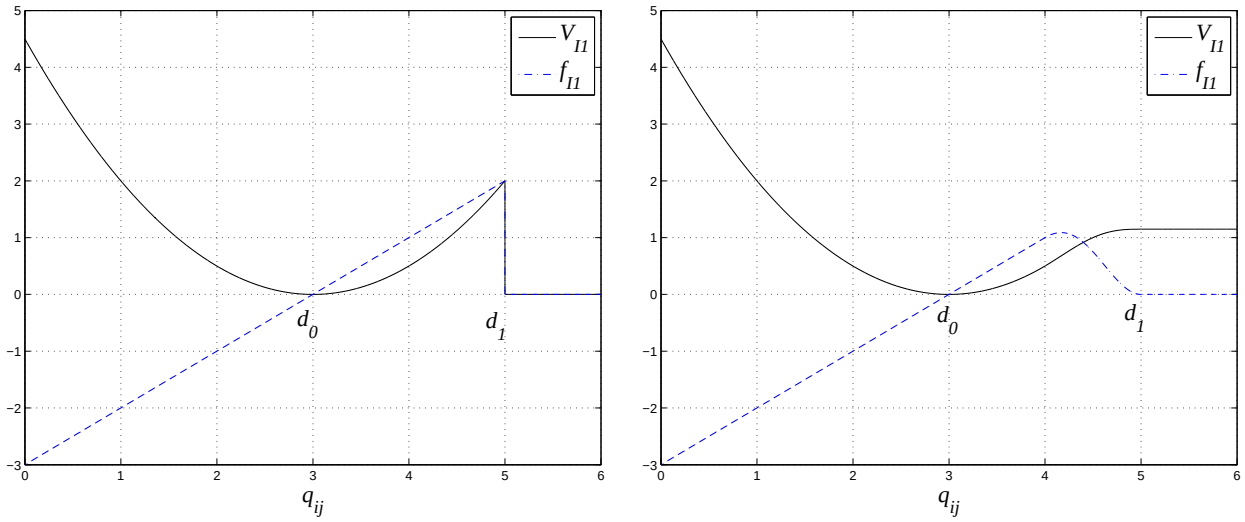
The force is defined by

$$\mathbf{F}_{I1}(\mathbf{q_i}) = -\nabla_{\mathbf{q_i}} V_{I1} = -f_{I1}(q_{ij})\hat{\mathbf{q}}_{\mathbf{ij}} = f_{I1}(q_{ij})\hat{\mathbf{q}}_{\mathbf{ji}} \tag{2.13}$$

with

$$f_{I1}(q_{ij}) = \begin{cases} \xi_I(q_{ij} - d_0) & 0 < q_{ij} \leq d_1 \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.14}$$

where we have explicitly defined the force at $q_{ij} = d_1$ (the discontinuity in V_{I1}). Figure 2.8(a) shows the plot of both functions V_{I1} and f_{I1}, were $\xi_I = 1$,$d_1 = 3$ and $d_1 = 5$.



(a) (b)

Figure 2.8: Vehicle-to-Vehicle Interaction Potentials. (a) Potential profile for V_{I1} and its gradient with $\xi_I = 1$, $d_1 = 3$, $d_1 = 5$. (b) Profile of numerically integrated potential whose gradient is given in 2.16, with $\xi_I = 1$, $d_0 = 3$, $d_1 = 5$, $a = 4$, and $b = 5$.

It can be observed that both functions are discontinuous at $q_{ij} = d_1$. To avoid this discontinuity, the force function is scaled by a bump function $\beta(q_{ij}) \in [0, 1]$, so that the resulting potential is at least C^1. An example of a bump function (so named for its shape) is suggested by Fiorelli in [3]. The function is given by equation (2.15) and is parameterized by a and b. Figure 2.9 shows the respective plot.

$$\beta(x) = \begin{cases} 1 & 0 < x \leq a \\ sin^2\left(\frac{\pi}{2}\frac{x-b}{a-b}\right) & a < x \leq b \\ 0 & x > b \end{cases} \tag{2.15}$$

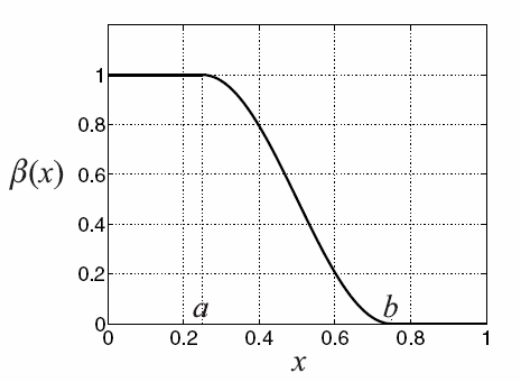


Figure 2.9: Bump Function. From [3]

The scaled force magnitude f_{I1} will then becomes

$$f_{I1} = \xi_I \beta(q_{ij}) \left(q_{ij} - d_0\right), \quad q_{ij} > 0 \tag{2.16}$$

Figure 2.8(b) shows the plot of the scaled vehicle-to-vehicle interaction potential V_{I1}, with the corresponding gradient f_{I1}, using the following parameters: $\alpha_I = 1$, $d_0 = 3$, $d_1 = 5$, $a = 4$ and $b = 5$.

Other Inter-vehicle Potential Functions

Inter-vehicle potentials can assume other profiles. Following the APF given by equation (2.3), the interaction potential V_{I2} can be given by

$$V_{I2}(\mathbf{q_{ij}}, \xi_I, d_0, d_1) = \begin{cases} \xi_I(q_{ij} - d_0) & 0 < q_{ij} \leq d_1 \\ \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.17}$$

with

$$\mathbf{F}_{I2}(\mathbf{q_i}) = \begin{cases} -\xi_I \hat{\mathbf{q}}_{\mathbf{ij}} & 0 < q_{ij} \leq d_0 \\ \xi_I \hat{\mathbf{q}}_{\mathbf{ij}} & d_0 < q_{ij} \leq d_1 \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.18}$$

$\mathbf{F}_{I2}$ presents a constant force magnitude for all $q_{ij} \neq d_0$. It is clearly not singular at d_0, and this fact may lead to instability problems in the formation causing oscillations. The application of this potential and these aspects are explored in Chapter 5. Figure 2.10 shows the profile of V_{I2} and its gradient, following the examples presented for V_{I1}. Figure 2.10(b) shows the scaled functions. Other functions can be adopted for

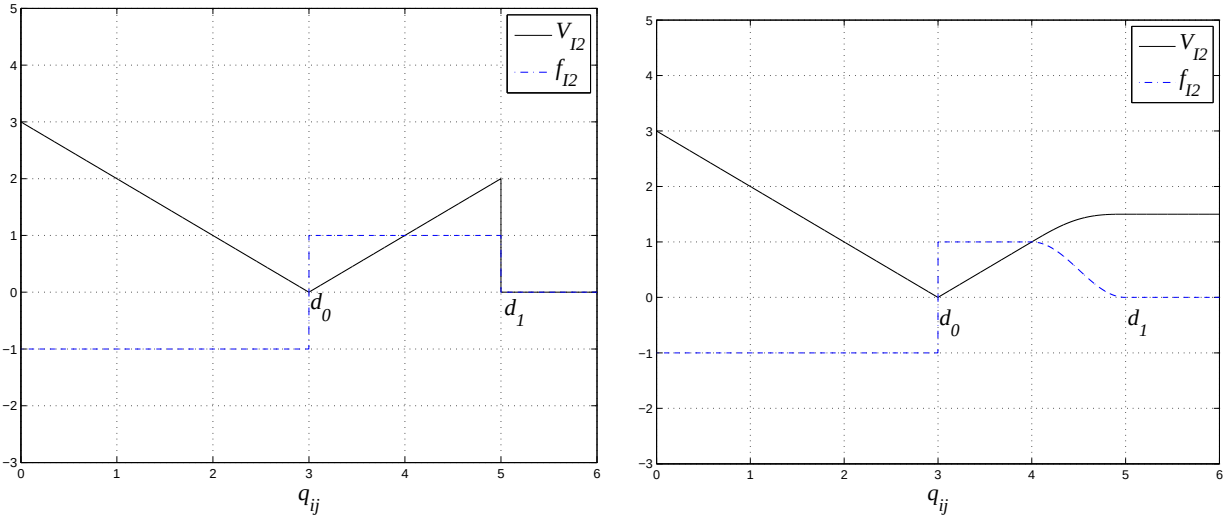


Figure 2.10: Vehicle-to-Vehicle Interaction Potentials. Profile of V_{I2} and its gradient, following the examples presented in figure 2.8.

the potential field equation. See, for example, the paper by Hettiarachchi and Spears [31]. Fiorelli and Leonard presented, in [3] and [16], the potential function that we adopted in most simulations of Chapters 4 and 5. For this particular potential, as $q_{ij} \to 0$, $f_{I3} \to \infty$, which is intended to prevent vehicle-to-vehicle collisions. The potential function is given by

$$V_{I3}(\mathbf{q_{ij}}, \xi_I, d_0, d_1) = \begin{cases} \vartheta & 0 < q_{ij} \leq d_1 \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.19}$$

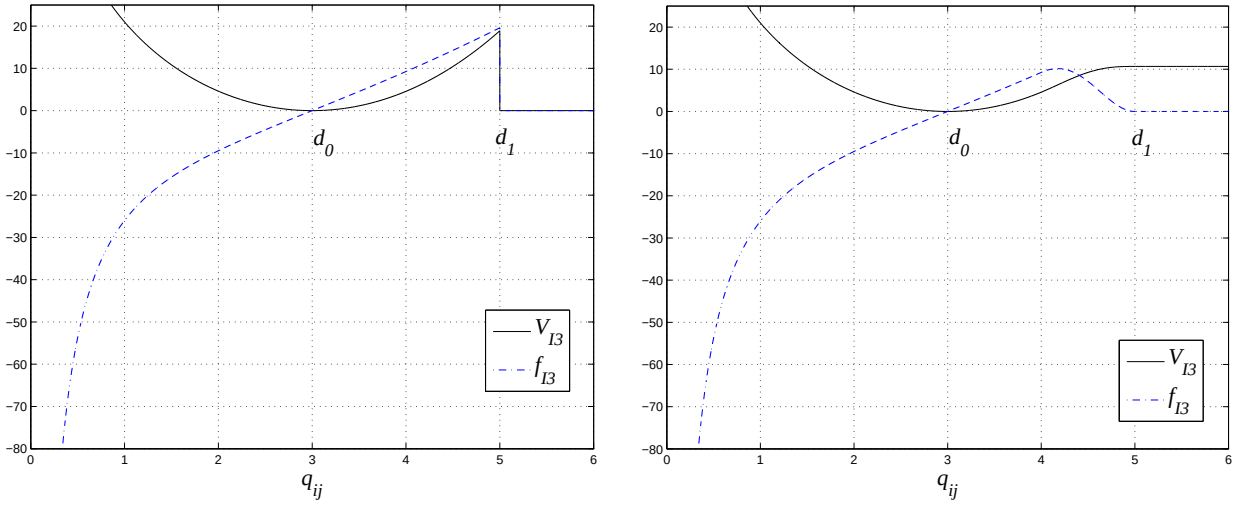
were

$$\vartheta = \xi_I \left(\frac{1}{3} q_{ij}^3 - d_0^3 ln(q_{ij}) - \frac{1}{3} d_0^3 + d_0^3 ln(d_0) \right)$$

and thus,

$$f_{I3}(\mathbf{q_i}) = \begin{cases} \xi_I \left(q_{ij}^2 - \frac{d_0^3}{q_{ij}} \right) & 0 < q_{ij} \leq d_1 \\ 0 & q_{ij} > d_1 \end{cases} \tag{2.20}$$

where $\mathbf{F}_{I3}(\mathbf{q_i})$ follows from equation (2.13) with f_{I3}. Figure 2.11 shows the Vehicle-to-Vehicle Interaction Potential V_{I3}, its gradient, and the scaled functions in figure 2.11(b), following the example given for the previous potentials V_{I1} and V_{I2}.



(a) (b)

Figure 2.11: Vehicle-to-Vehicle Interaction Potentials. Profile of V_{I3} and its gradient, following the examples presented in figures 2.8 and 2.10.

2.3 Moving the Formation

Centralized architectures are characterized by a single control agent, as opposed to a fully decentralized approach that lack such an agent. The behavior of decentralized systems is often described using such terms as emergence and self-organization, and it is widely claimed that these architectures include fault tolerance, natural parallelism, reliability and scalability [32]. This conforms with the proposed model at the level were each agent manages its one path within the FF. The presented potentials guarantee the necessary inter-vehicle spacing and group cohesion. When a team is not moving, the formation will converge to a static flock formation pattern that depends on the number of robots. This configuration, that is stretchable when the formation is in translation, corresponds to a local minimum in the overall potential fields. Figure 2.12 shows an example of a stabilized formation with 7 robots and the respective inter-vehicle connections. For some applications, it may be desirable to include some random movement in each robot to generate effects similar to insect swarms. This can be done by adding a random force vector to equation (2.11). The effect will animate the formation over a specified area, maintaining at the same time group cohesion and avoiding inter-vehicle collisions. This can be used, *e.g.*, for area exploration and scientific mapping.

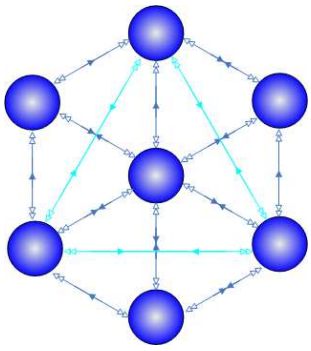


Figure 2.12: Stabilized Formation at equilibrium point and Forces.

A level of centralization is needed to set a mission goal or destination. Formation translation and deployment is achieved by moving a member of the team. This member, selected as the team *Virtual Leader* (VL), will in turn "drag" the rest of the formation behind by taking the formation out off equilibrium. When the VL shifts from its equilibrium position the entire flock will follow towards a new equilibrium configuration. The advantage of this methodology is that only a single trajectory (the leaders trajectory) needs to be planned. The adopted formation methodology will take care of resolving the collision free paths for each member of the team. Figure 2.13 shows a selected leader moving with the rest of the formation.

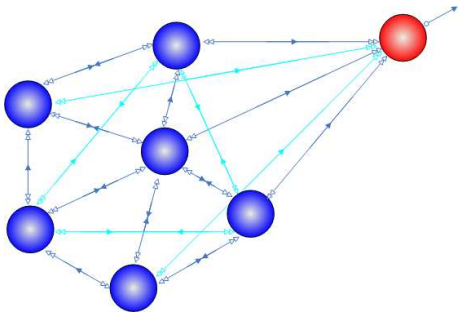


Figure 2.13: Moving the formation by dragging a selected leader (in red).

The selection of the leader may depend on some mission dependent criteria, that must be applied every time a problem or malfunction happens to the current leader. To avoid this election process, instead of using a team robot for this task, a virtual reference point can be used to guide the team. This imaginary point will have an associated dynamic and will be included as a valid configuration in the formation.

The VL term was adopted from Fiorelli and Leonard [3],[16], but it is used in a

more relaxed fashion in the sense that only the inter-vehicle distance is regulated. There has been no need for any specific imposition regarding the orientation, or angular position, that a specific member should assume *w.r.t.* the VL.

2.4 Using Non-Holonomic Vehicles

It has been assumed up to now a direct application of the control input, given by equation (2.11), without any non-holonomic restrictions. The point of application, as shown in figure 2.14, is at the center position of a holonomic vehicle (see [33]). To generalize the concepts, and open way for heterogenous formations, an interest-

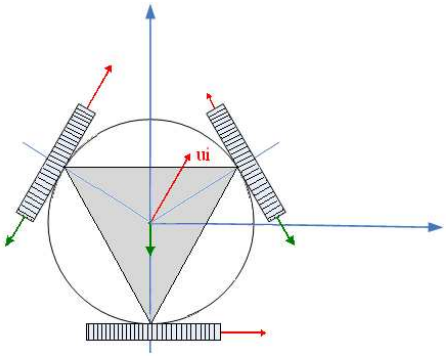


Figure 2.14: Controlling a holonomic vehicle.

ing extension of the ideas would be their application to non-holonomic vehicles. We reinforce this fact with the idea that these vehicles present several advantages over omnidirectional drive, when operating in rough outdoor terrains. The limitation that arises when directly including them in the formation framework (FF), is that its not possible to control and stabilize the position plus orientation of these vehicles with a time-invariant, stabilizing control strategy (see [34],[35] and [36]). The control input cannot be directly applied to the center of the vehicle as it is done with holonomic vehicles. It is known, however, that for the differential-drive class of non-holonomic wheeled vehicles, control forces can be applied on a off-wheel axis point, whose kinematics can be made holonomic by using a suitable transformation [37], [38],[39],[40]. This way, the formation control algorithms described in the previous section can be applied to non-holonomic vehicles. Figure 2.15 shows the *Holonomic Point* (HP) on a non-holonomic differentially driven vehicle.

The HP position $\mathbf{q_i} = (h_x, h_y)^T$ lies at a distance $L = \alpha.R$ along a line that is normal to the wheel axis, intersecting the axis at the robots center point $\mathbf{r} = (r_x, r_y)^T$,

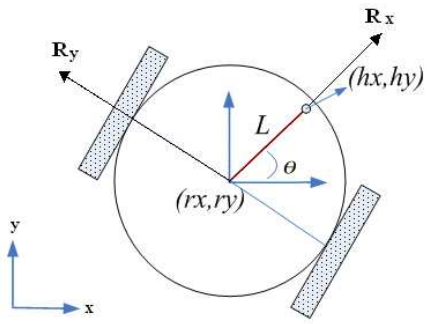


Figure 2.15: World and robot frames, and the off-wheel holonomic point (h_x, h_y) on a non-holonomic differentially driven vehicle.

where R is the robots radius and $\alpha \in]0, 1]$. The kinematics of the HP position is holonomic for $L \neq 0$.

The differential driven vehicle has the following equations of motion:

$$\begin{pmatrix} \dot{r}_{xi} \\ \dot{r}_{yi} \\ \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \end{pmatrix} = \begin{pmatrix} v_i \cos(\theta_i) \\ v_i \sin(\theta_i) \\ \omega_i \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_i} & 0 \\ 0 & \frac{1}{J_i} \end{pmatrix} \begin{pmatrix} \mathbf{F}_i \\ \tau_i \end{pmatrix} \tag{2.21}$$

where $\mathbf{r}_i = (r_{xi}, r_{yi})^T$ is the inertial position of the ith robot, θ_i is the orientation, v_i is the linear speed in the robot frame, ω_i is the angular speed, τ_i is the applied torque, F_i is the applied force, m_i is the mass, and J_i is the moment of inertia. Letting $\mathbf{x}_i = (r_{xi}, r_{yi}, \theta_i, v_i, \omega_i)$, and $\mathbf{u}_i = (F_i, \tau_i)$, the equations of motion can be written as

$$\dot{\mathbf{x}}_i = f(\mathbf{x}_i) + g_i \mathbf{u}_i \tag{2.22}$$

where the definitions of f and g_i can be inferred from (2.21). The configuration vector $\mathbf{q_i}$ and the velocity vector $\dot{\mathbf{q}}_\mathbf{i}$, associated with the HP, are given by the following equations:

$$\mathbf{q_i} = \mathbf{r}_i + L_i \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} \tag{2.23}$$

$$\dot{\mathbf{q}}_\mathbf{i} = \dot{\mathbf{r}}_\mathbf{i} + L_i \frac{\partial}{\partial t} \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) & -L_i \sin(\theta_i) \\ \sin(\theta_i) & L_i \cos(\theta_i) \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} \tag{2.24}$$

Differentiating 2.24 with respect to time gives

$$\ddot{\mathbf{q}}_\mathbf{i} = \begin{pmatrix} \dot{v}_i\cos(\theta_i) - v_i\omega_i\sin(\theta_i) - \dot{\omega}_i L_i\sin(\theta_i) - L_i\omega_i^2\cos(\theta_i) \\ \dot{v}_i\sin(\theta_i) + v_i\omega_i\cos(\theta_i) + \dot{\omega}_i L_i\cos(\theta_i) - L_i\omega_i^2\sin(\theta_i) \end{pmatrix}$$

equal to

$$\ddot{\mathbf{q}}_\mathbf{i} = \begin{pmatrix} -v_i\omega_i\sin(\theta_i) - L_i\omega_i^2\cos(\theta_i) \\ v_i\omega_i\cos(\theta_i) - L_i\omega_i^2\sin(\theta_i) \end{pmatrix} + \begin{pmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{pmatrix} \begin{pmatrix} F_i \\ \tau_i \end{pmatrix}$$

We can define a mapping $\Psi(x_i) : \mathbb{R}^5 \to \mathbb{R}^5$ between the robots state vector and a state vector associated with the HP position.

$$\delta_i = \Psi(x_i) \triangleq \begin{pmatrix} r_{xi} + L_i\cos(\theta_i) \\ r_{yi} + L_i\sin(\theta_i) \\ v_i\cos(\theta_i) - L_i w_i\sin(\theta_i) \\ v_i\sin(\theta_i) + L_i w_i\cos(\theta_i) \\ \theta_i \end{pmatrix} \tag{2.25}$$

$$\delta_i = (\delta_{1i}, \delta_{2i}, \delta_{3i}, \delta_{4i}, \delta_{5i})^T$$

$$\mathbf{q_i} = \begin{pmatrix} \delta_{1i} \\ \delta_{2i} \end{pmatrix} \quad \text{and} \quad \dot{\mathbf{q}}_\mathbf{i} = \begin{pmatrix} \delta_{3i} \\ \delta_{4i} \end{pmatrix}$$

The map Ψ is a diffeomorphism, and its inverse is given by $x_i = \Psi^{-1}(\delta_i)$

$$\begin{aligned} \delta_{3i} &= v_i\cos(\delta_{5i}) - L_i\omega_i\sin(\delta_{5i}) \\ \delta_{4i} &= v_i\sin(\delta_{5i}) + L_i\omega_i\cos(\delta_{5i}) \end{aligned}$$

from δ_{3i}

$$v_i = \frac{\delta_{3i} + L_i\omega_i\sin(\delta_{5i})}{\cos(\delta_{5i})}$$

placing v_i in δ_{4i} we have

$$\begin{aligned} & \delta_{4i} = \left(\tfrac{\delta_{3i} + L_i\omega_i\sin(\delta_{5i})}{\cos(\delta_{5i})}\right)\sin(\delta_{5i}) + L_i\omega_i\cos(\delta_{5i}) \\ \equiv\ & \delta_{4i}\cos(\delta_{5i}) = \delta_{3i}\sin(\delta_{5i}) + L_i\omega_i\sin^2(\delta_{5i}) + L_i\omega_i\cos^2(\delta_{5i}) \\ & \delta_{4i}\cos(\delta_{5i}) - \delta_{3i}\sin(\delta_{5i}) = L_i\omega_i\left[\sin^2(\delta_{5i}) + cos^2(\delta_{5i})\right] \end{aligned}$$

$$\omega_i = -\frac{1}{L_i}\delta_{3i}\sin(\delta_{5i}) + \frac{1}{L_i}\delta_{4i}\cos(\delta_{5i}) \tag{2.26}$$

placing ω_i in δ_{3i} we have

$$\begin{aligned}
& \delta_{3i} = v_i \cos(\delta_{5i}) - L_i \left[-\tfrac{1}{L_i}\delta_{3i} \sin(\delta_{5i}) + \tfrac{1}{L_i}\delta_{4i} \cos(\delta_{5i}) \right] \sin(\delta_{5i}) \\
\equiv \quad & \delta_{3i} = v_i \cos(\delta_{5i}) + \delta_{3i} \sin^2(\delta_{5i}) - \delta_{4i} \cos(\delta_{5i}) \sin(\delta_{5i}) \\
& \delta_{3i} \left(1 - \sin^2(\delta_{5i})\right) + \delta_{4i} \cos(\delta_{5i}) \sin(\delta_{5i}) = v_i \cos(\delta_{5i}) \\
& \delta_{3i} \cos^2(\delta_{5i}) + \delta_{4i} \cos(\delta_{5i}) \sin(\delta_{5i}) = v_i \cos(\delta_{5i})
\end{aligned}$$

$$v_i = \delta_{3i} \cos(\delta_{5i}) + \delta_{4i} \sin(\delta_{5i}) \tag{2.27}$$

and so $\Psi^{-1}(\delta_i)$ is given by

$$\Psi^{-1}(\delta_i) = \begin{pmatrix} \delta_{1i} - L_i \cos(\delta_{5i}) \\ \delta_{2i} - L_i \sin(\delta_{5i}) \\ \delta_{5i} \\ \delta_{3i} \cos(\delta_{5i}) + \delta_{4i} \sin(\delta_{5i}) \\ -\frac{1}{L_i}\delta_{3i} \sin(\delta_{5i}) + \frac{1}{L_i}\delta_{4i} \cos(\delta_{5i}) \end{pmatrix}$$

In the transformed coordinates, equation (2.22) is given by

$$\begin{aligned}
\begin{pmatrix} \dot{\delta}_{1i} \\ \dot{\delta}_{2i} \end{pmatrix} &= \begin{pmatrix} \dot{\delta}_{3i} \\ \dot{\delta}_{4i} \end{pmatrix} \\
\begin{pmatrix} \dot{\delta}_{3i} \\ \dot{\delta}_{4i} \end{pmatrix} &= \begin{pmatrix} -v_i\omega_i \sin(\theta_i) - L_i\omega_i^2 \cos(\theta_i) \\ v_i\omega_i \cos(\theta_i) - L_i\omega_i^2 \sin(\theta_i) \end{pmatrix} + \begin{pmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{pmatrix} \mathbf{u}_i \\
\dot{\delta}_{5i} &= -\tfrac{1}{L_i}\delta_{3i} \sin(\delta_{5i}) + \tfrac{1}{L_i}\delta_{4i} \cos(\delta_{5i})
\end{aligned}$$

Given that the following matrix is singular

$$A = \begin{pmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{L_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & \frac{L_i}{J_i}\cos(\theta_i) \end{pmatrix}, \qquad detA = \frac{L_i}{m_i J_i} \neq 0$$

an output feedback linearizing control can be stipulated as:

$$\mathbf{u}_i = A^{-1} \left[\mathbf{c}_i - \begin{pmatrix} -v_i\omega_i \sin(\theta_i) - L_i\omega_i^2 \cos(\theta_i) \\ v_i\omega_i \cos(\theta_i) - L_i\omega_i^2 \sin(\theta_i) \end{pmatrix} \right] \tag{2.28}$$

which gives the following linear state equation

$$\begin{aligned}
\begin{pmatrix} \dot{\delta}_{1i} \\ \dot{\delta}_{2i} \end{pmatrix} &= \begin{pmatrix} \delta_{3i} \\ \delta_{4i} \end{pmatrix} \\
\begin{pmatrix} \dot{\delta}_{3i} \\ \dot{\delta}_{4i} \end{pmatrix} &= \mathbf{c}_i \\
\dot{\delta}_{5i} &= -\tfrac{1}{L_i}\delta_{3i} \sin(\delta_{5i}) + \tfrac{1}{L_i}\delta_{4i} \cos(\delta_{5i})
\end{aligned}$$

with $\mathbf{c}_i$ being the control input in the transformed coordinates.

A limitation in this procedure is the fact that the transformation will render the robots orientation uncontrollable. This is not a limitation to the proposed objectives of having to deploy the robots to a specified destination. While in translation, the robots orientation will tend to become aligned with the velocity vector applied to the HP. Figure 2.16 shows an example of a differential drive robot drawing a red square with a holonomic point placed on its front end. The black lines are the traces of each wheel axis along the entire drawing process.

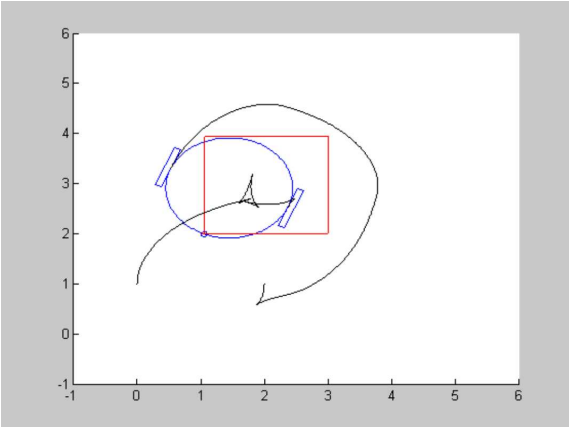


Figure 2.16: The track of an Holonomic Point placed on the front end of a differential drive robot (in red). The black lines represent wheel tracks.

2.5 Scalability and Hierarchical Organization

In engineering, scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, or to be readily enlarged. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources (typically hardware) are added. In our case, the definition aims directly to the framework structure and performance when more robots are added. Although, until now, no mention has been made to the underling communication network that may be needed to support the formation, a natural concern arises when we consider larger formations and their impact on the performance of this network and on the necessary information flow. Scalability, as a property of the formation, would be the capacity to increase performance (that depends on the mission) with the increase of the number of ele-

ments deployed out to the mission. In the present FF configuration every element, excluding the leader, will consider the configuration of all the other members. An alternative and more scalable solution would be, for example, the organization of a FF into several clusters $(C_1, \ldots, C_n)$, with each cluster composed by several groups $(G_1, \ldots, G_n)$ as shown in figures 2.17 and 2.18.

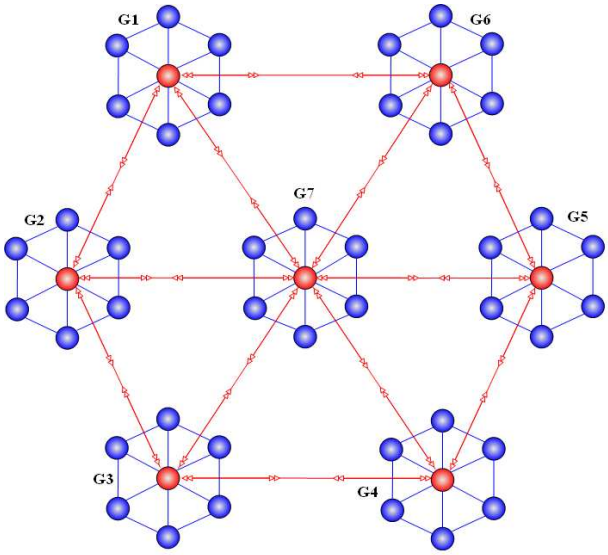


Figure 2.17: FF Hierarchical Organization - a cluster C_i, composed by 7 groups.

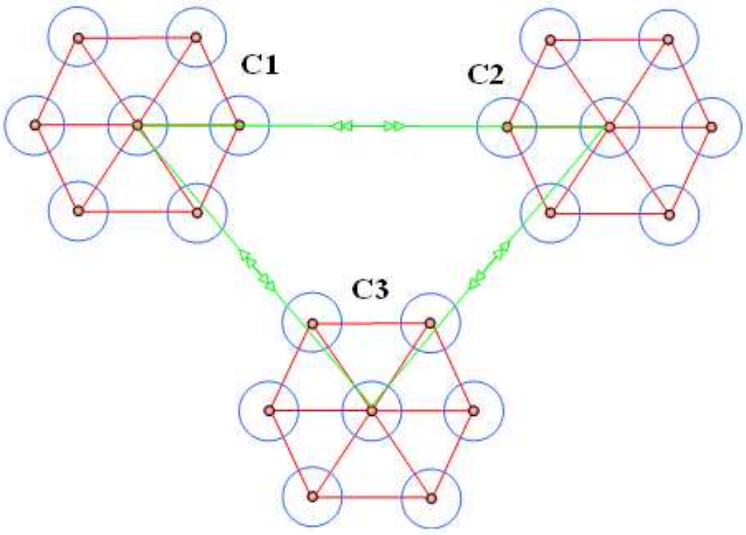


Figure 2.18: A FF with 3 Clusters.

Each group has a selected leader that takes part of higher framework organization

with independent IPs. The interactions are confined to each group and the solution can be scaled to higher hierarchical levels. Instead of knowing the configuration of every member of the formation, vehicles will only consider the other members of its group, reducing the necessary information exchange among vehicles. A concluding remark on this solution calls for some attention on the selected inter-vehicle distances for each hierarchical level, to guarantee the necessary spacing between groups and clusters and avoid inter-vehicle collisions.

3

Obstacle Avoidance

Navigation is the problem of finding a collision-free motion for the robot system from one configuration (or state) to another. [41]

When a team of robots moves in an uncontrolled and unknown environment the problem of obstacle avoidance must inevitably be considered. This assumes particular importance in the present architecture that intends to alleviate the task of having an operator defining individual paths for each element of the formation. Each vehicle should move through space and avoid dynamic, static, known and unknown obstacles and forbidden areas. In this chapter some solutions to this "behavior" are presented and discussed. It will be assumed that each robot will sense obstacles within a limited range from its current position, and may not have any other knowledge of the surrounding environment and world map. The first section presents the obstacle avoidance algorithm using repulsive APFs. A discussion of some limitations due to local minimum situations in PFM will follow, with the presentation of some solutions to the problem.

3.1 Obstacle Detection and Avoidance

The purpose obstacle detection is to identify, for a specified robot i, the set of obstacles $\mathbf{o_i} \in O$ that exist within a certain range of the area defined by $A(\mathbf{q_i})$. Several systems are used in practical applications that include, *e.g.*, ultrasound and infrared sensor rings, laser range finders, cameras and others like radar-based obstacle detection systems, that are emerging in the automotive industry. Each system will present several particular aspects and limitations to be considered when used. For the purpose of the obstacle avoidance algorithms presented in this chapter, it can even be assumed that an external source will stream the necessary information $\mathbf{o_i}$ to the formation. From the point of view of these algorithms an *Obstacle Detection Layer* (ODL) will supply a set of discrete readings $[d_0 \dots d_z]$, that correspond to the distances between points placed on the robots boundary at the respective radial bearings $[\theta_1 \dots \theta_z]$, and the nearest points of an obstacle, or obstacles, over straight lines with such orientation, as shown in figure 3.1.

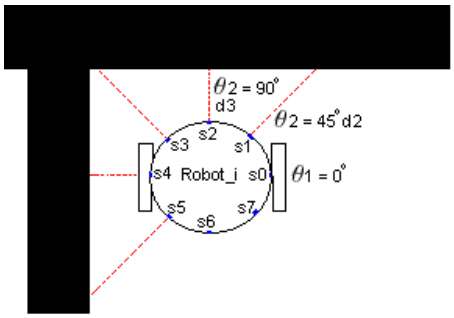


Figure 3.1: Data obtained from the *Obstacle Detection Layer*.

To avoid obstacles, a repulsive force must be added with the formation forces to repel each robot from the obstacles. The readings obtained from the detection layer will generate repulsive forces and the total force is given by:

$$\mathbf{F}_{rep}(\mathbf{q_i}) = \sum_{i=1}^{z} \mathbf{F}_{rep,\theta_i}(\mathbf{q_i}) \tag{3.1}$$

where $\mathbf{F}_{rep,\theta_i}(\mathbf{q_i})$ is a repulsive force associated to the value d_i measured in configuration $\mathbf{q_i}$, at the angle θ_i. Following the potential function and repulsive force given by equations (2.8) and (2.9), the repulsive potential is given by:

$$U_{rep,\theta_i}(\mathbf{q_i}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d_i} - \frac{1}{d_l}\right)^2 & d_i \leq d_l \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

and we have

$$\mathbf{F}_{rep,\theta_i}(\mathbf{q_i}) = \begin{cases} \eta\left(\frac{1}{d_i} - \frac{1}{d_l}\right)\left(\frac{1}{d_i^2}\right)\hat{\mathbf{e}}_\mathbf{i} & d_i \leq d_l \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $\hat{\mathbf{e}}_\mathbf{i} = (1, \theta_i \pm \pi)$ is the unit vector given in polar coordinates, that indicates the direction of the repulsive force.

The control force given by equation (2.11) will now become:

$$\mathbf{u_i} = -\nabla_{\mathbf{q_i}} V_i - \mathbf{f_{vi}} - \mathbf{F}_{rep}(\mathbf{q_i}) \tag{3.4}$$

3.2 Local minima

Originally proposed and well-suited for on-line planning where obstacles are "sensed" during motion execution, the PFM present a well known problem associated with the possible existence of local minima situations, that can get robots blocked and trapped in certain configurations. This occurs very frequently behind obstacles where repulsive forces counterbalance the attraction forces. To deploy the formation to a target area, a local minima free path will be needed to guide the leader throughout the obstacle field. The *Interaction Forces* (IFs) between all the robots will "drag" the formation behind and in some situations, robots can get stuck in a local minima configuration if the leader is drawn away and around an obstacle. Figure 3.2 shows an example of this situation with the leader leaving a red track and each one of the other three robots, a black track.

To minimize these occurrences the leader will have to check, while moving, the formation status and avoid getting ahead in such a way that the robots are directed towards the obstacles, instead of being guided around them. The leader must be kept forcedly close to guarantee this requirement. The desired behavior for each robot is to follow the leader while "flowing" smoothly and unattended around the obstacles, without getting trapped. This way the leader can travel freely without any constraints, and be guided directly to the target area. The following sections will focus on this subject and discuss some solutions to this problem.

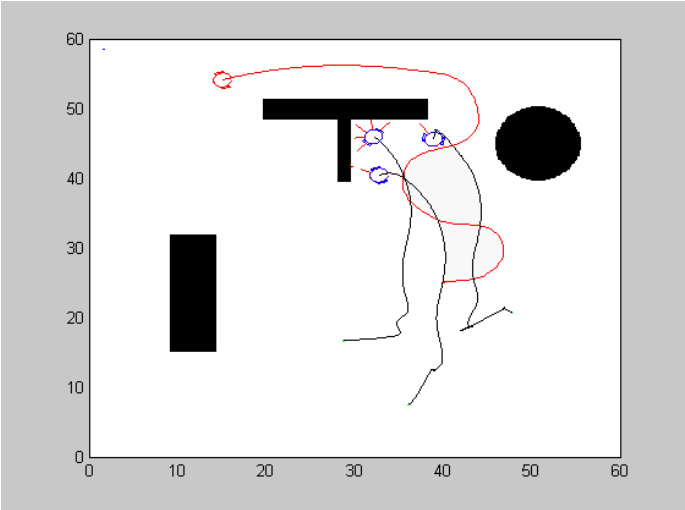


Figure 3.2: Team getting stuck in a local minima while trying to follow the leader.

3.3 Following the Leader's Track

3.3.1 Searching For Clearance

The first step to avoid falling into a local minima situation is detecting this condition. Subsequently, a natural heuristic is to focus on the path followed by the leader around the obstacle. This information can be available if the robot keeps a **track log** of the last n positions assumed by the leader $\mathbf{Q}_{LOG} = [\mathbf{q_l}(t), \mathbf{q_l}(t-1), \ldots, \mathbf{q_l}(t-n)]$ for t and n discrete . As soon as a robot senses a repulsive force that tends to break the formation with the leader, the robot must search for another reference point to replace the leader's current position $\mathbf{q_l}(t)$. In this case, the robot searches back the track log for a suitable position goal that avoids the obstacle, starting at the leaders current position. This behavior is selected from the continuous analysis of the resulting repulsion force vectors $\mathbf{F}_{rep}(\mathbf{q_i})$ generated by the obstacles, as explained in the sequel. To keep following the leader the robot needs a minimum clearance area between $\mathbf{q_i}$ and $\mathbf{q_l}$. This area corresponds to a cone that connects both robots, as the one represented in front of the robot towards the leader, in Figure 3.3.

Clearance is validated by checking the dot product between $\mathbf{F}_{rep}(\mathbf{q_i})$ and $\hat{\mathbf{q}}_{\mathbf{li}}$, with $\mathbf{F}_{rep}(\mathbf{q_i}) \neq 0$. The following condition has to be verified:

$$\mathbf{F}_{rep}(\mathbf{q_i}) \cdot \hat{\mathbf{q}}_{\mathbf{li}} > \xi_{max} \tag{3.5}$$

where ξ_{max} is a simulation parameter and will correspond to the value of the dot

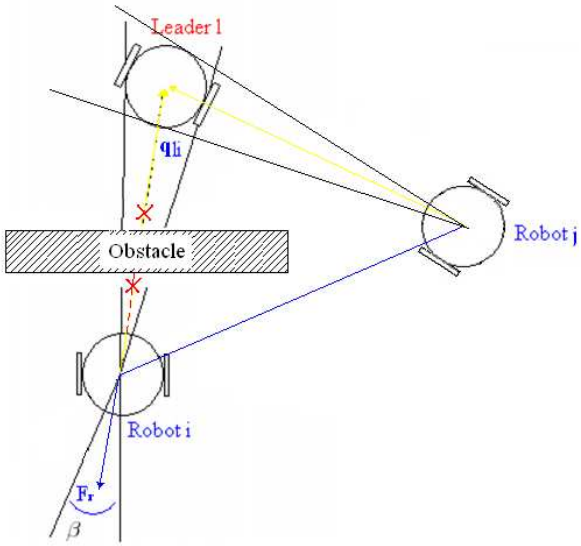


Figure 3.3: Robot clearance detection.

product between $\hat{\mathbf{q}}_{\mathbf{li}}$ and a unit vector that falls on the edge of the cone represented in figure 3.3 and figure 3.4, with β representing its aperture. The vehicle will consider having no clearance if the repulsive vector falls inside this cone, *i.e.*, $\mathbf{F}_{rep}(\mathbf{q_i}) \cdot \hat{\mathbf{q}}_{\mathbf{li}} \in [-1, \xi_{max}]$.

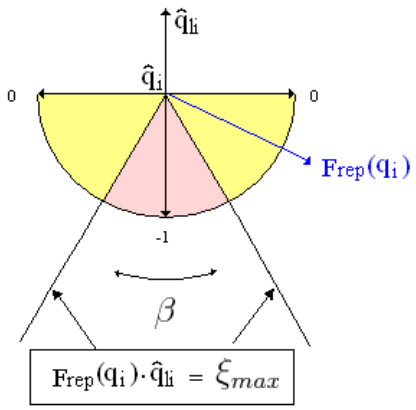


Figure 3.4: Clearance detection condition.

If the robot does not have clearance to follow the leader's current position it must find another suitable point, in $\mathbf{Q}_{LOG}$, to replace that position in the formation framework. Algorithm 1 summarizes the idea and figure 3.5 contains a representation

of the procedure, with blue lines representing links to points that have no clearance and a yellow line, representing a link to the *Current Attraction Point* (CAP), *i.e.*, the first point that verifies the condition.

Algorithm 1: Tracking the Leader.

$Q_{LOG}(headPosition)$ = LeaderCurrentPos
for *posIndex = headPosition until endOf*(Q_{LOG}) **do**
    currAttractionPoint = $Q_{LOG}(posIndex)$
    if *clearanceCond(currAttractionPoint) == true* **then**
        modifySimulationParameters
        break and use currAttractionPoint

If the robot is tracking the leader and the CAP corresponds to a point in Q_{LOG} either than the leader's current position, then the IP parameters are rearranged. The distance of equilibrium d_0 is altered to 0 to follow on the tracking path, and the magnitude of attraction is adapted to accelerate the tracking process. The values are then restored to their original settings as soon as the robot regains clearance to the leader.

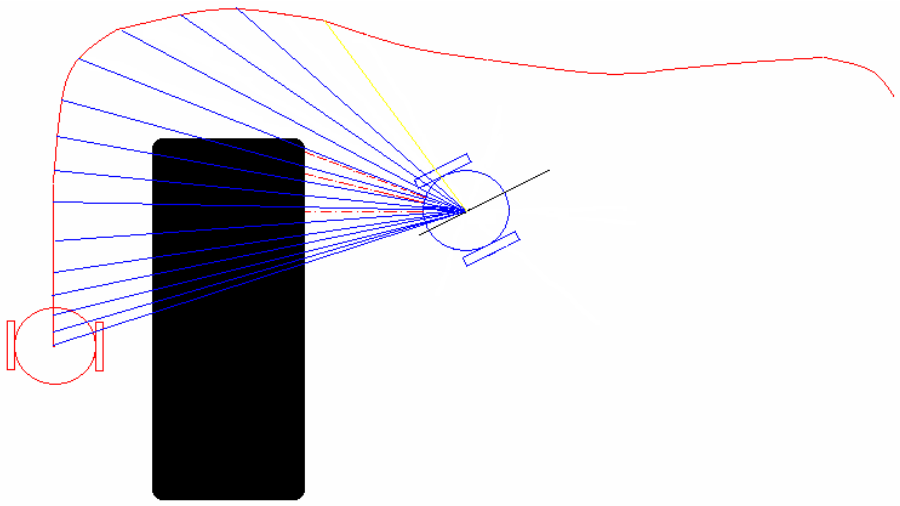


Figure 3.5: The leader tracking procedure.

3.3.2 Breaking The Formation

In order to maintain group cohesion actions may be taken that lead the robots to get stuck in the awkward situation of being unable to follow the leader due to the interactions with other teammates that got stuck behind. In some situations, a group of robots way even be malfunctioning or in some condition that may be affecting negatively the performance of the whole formation. For this and other possible reasons, dynamic link breaking and creating is a topic to be explored in future work. For now, and to facilitate the "formation flow" around the obstacles, a robot i breaks the formation link with any teammate j, if $\mathbf{F}_{rep}(\mathbf{q_i})$ and $\hat{\mathbf{q}}_{\mathbf{ji}}$ fail to verify the clearance condition. Figure 3.6 shows an example. With this procedure some robots may become isolated from the rest of the formation. However, it should be noted that all robots will try and keep up with the leader and in consequence, broken links will be reactivated as soon as the robots regain clearance among themselves.

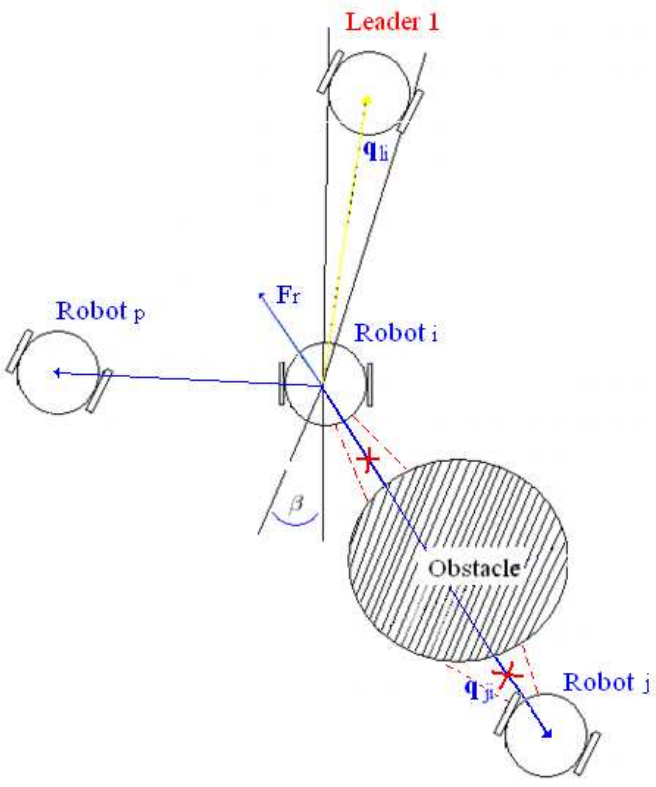


Figure 3.6: Breaking a formation link with robot j.

3.4 Discussion

For every if-else and loop statement included in the leader-following code, the implementation drifts further and further away from the initial quest of keeping the reactive behaviors as simple as possible. Ideally, the type of solution that would fit the current line of work would be to use, just like the PFM approach, a closed form mathematical expression to generate the necessary force vectors that would lead the vehicles to their target without getting trapped behind obstacles. There is some knowledge in the formation that can be useful to help the robots avoid local minimum situations. Each robot has full knowledge of the position of all the other teammates and the disposition of the formation, combined with the path followed by the leader, may provide hints about the geometry of the free-space and obstacles. The overall group knowledge about the surrounding world can further be extended with distributed cooperative sensing and other information exchange, that may help the whole formation flow around the obstacles without getting trapped. Ideally, to support the original quest, the desired result would be a force vector to guide each robot to its destination, generated with "low" computational complexity from a model of the free-space created from all the available information. Appendix A presents *Navigation Functions* (NF) as an approach considered for this effect. To build such functions a prior global knowledge of the entire region must be available. Therefore, some ideas will be discussed for future work, in Chapter 6, on how to use these methods with the knowledge available in the formation.

4

Simulation Setup

This chapter describes the set of experiments performed in a simulated environment to test and evaluate the algorithms presented in the former chapters. It is well known that simulation is particularly useful for preliminary studies before the final implementation of the algorithms on real robots. This takes particular importance when multiple robots are involved, and were setting up a homogeneous team of robots is quite expensive and time consuming. With simulation, multiple scenarios can be presented, that may also include uncommon and rare situations difficult to model and verify in a more practical environment. The advantages do not come without cost because simulation errors exist, and practical implementation is, by itself, a whole new problem that may even render the theoretical project unpractical. The risk becomes even more present when strong assumptions are made like, for example, assuming that each robot knows his exact position and of all his teammates, operates in a world with perfect radio communications, the obstacle detection layer will function perfectly as described in Chapter 3, and no problems will arise due to some other aspects like, for example, friction, slippage, acceleration and velocity limits, *etc.* But even with these limitations, the presented simulated results help bring a better understanding

of the algorithms and opens way for future practical experimentations. The set of simulations presented in this chapter were performed using the Matlab environment, that was chosen as a first step towards the simulation of the algorithms before using other simulation platforms. The next section describes the simulation setup and, the subsequent sections, the experiments performed over the formation framework structure and the obstacle avoidance algorithms.

4.1 Matlab Simulation

Matlab provides a useful programming environment to explore and understand the theoretical aspects of the algorithms. The simulation setup presents a *virtual formation* composed by a leader and a set of robots that operates in a bounded world, modeled for some particular experiment. The leader, and subsequently the rest of the formation, is controlled by a remote station as shown in figure 4.1.

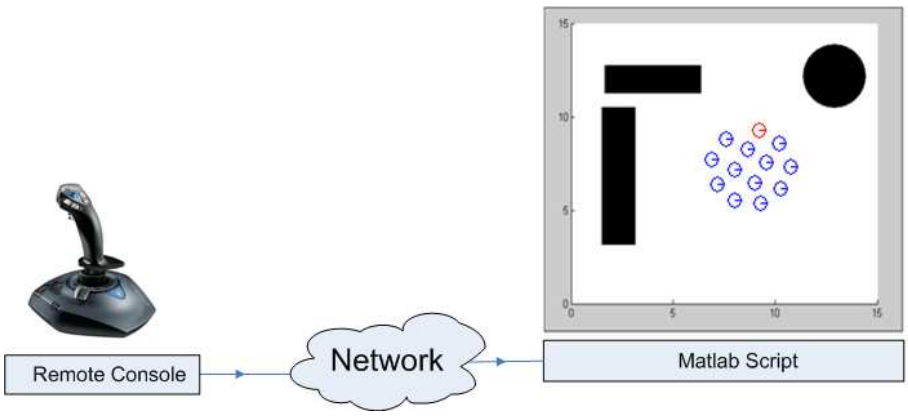


Figure 4.1: In the Simulation Setup a remote operator controls a virtual formation, using Matlab for the simulation.

The remote control station runs a console application. This program consists of a network client application, written in `C++`, that connects to a remote server and sends a stream of packed messages containing the necessary control instructions for the formation leader. In the experiments, the remote operator uses a joystick control interface to control the movements of a formation leader. Messages contain the current state of the joystick that includes, *e.g.*, the list of buttons that are currently being pressed and the configuration of the X-Y axis. The messages are processed by the server application, that was written as a Matlab function and executes in the

Matlab environment. The function shows the output simulation results on a *Graphical User Interface* using Matlab's plot tool. The graphical window shows the formation responding in real-time to commands sent by the operator, the bounding world, and additional information like, *e.g.*, lines representing formation forces, sonar obstacle detection, repulsion forces and information for debugging purposes.

In each simulation the representation of the world and the initial position of each element of the team is loaded from a bitmap file ("`world.bmp`"). The file contains obstacles represented by "black objects", the initial position of the leader represented by a red pixel and the initial positions of all the other members of the formation represented by green pixels. New elements can be added or removed from the formation just by editing this file and by adding or removing these pixels.

4.1.1 Matlab Formation Simulation Function

The simulation program operates in discrete time approximation to the continuous behavior of the robots, with a specified time-step δ_t. At each iteration, the state of each element of the formation is calculated in accordance with the previously presented equations, and then updated and displayed on the graphical output. Algorithm 2 summarizes the steps of the simulation program in its most basic form. The state variables that are associated to each robot are updated in different ways, depending on if the robots are holonomic or non-holonomic. A `DrawWorld` function displays the arena with the obstacles and, distinctive functions are used for drawing the graphical representation of the leader and all the other robots, from their current configurations. The position of each robot is logged so that tracks can be drawn and shown.

4.2 Formation Framework Experiments

In the initial set of experiments, from Test 1 to Test 7, the FF structures presented in Chapter 2 are simulated and evaluated. The dynamic and static behavior of the formation is analyzed for teams with different sizes, different potential scalings, and using holonomic and non-holonomic robots. Each test may contain one or more tables with the associated configuration parameters.

Brief description of each test:

Algorithm 2: Matlab Simulation Program

```
Initialize Simulation Parameters
Read World.bmp and Setup World Configuration
Setup Remote Communication Interface
Wait for Incoming Connection
Start Timer
while TerminationCondition = false do
    Read and Decode Control Messages from Remote Console
    Process Control Message:
    begin
        Verify Termination Condition
        Calculate Leader Velocities
    end
    Calculate Elapsed time δ_t
    Update Leader State Variables using δ_t
    Update State Variables of all other robots:
    begin
        for each robot of the team do
            Calculate the Interaction Force with the Leader (IF)
            for all other robots of the team do
                Update IF with the Force from the Interaction with this Robot
            Add and additional dissipative force f_vi to IF
            Update the robots State Variables using IF and δ_t
    end
    Draw World
    Draw Leader
    Draw the Rest of the team
    Update Logs
Close Connection with Remote Interface
```

- **Test 1** In this test the presented IPs V_{I1},V_{I2} and V_{I3} are simulated using holonomic robots.

- **Test 2** The objective is to verify the configuration that a certain formation will assume, if parameter d_1 is modified so that the distance of influence of

the leader-vehicle IP is significantly higher that the distance of influence of the vehicle-vehicle IP, breaking some formation links.

- **Test 3** In this test individual tracks are added to each vehicle to verify how vehicles move.

- **Test 4** Using 3 robots, the objective is to draw the individual paths followed by each one of these robots within a certain time frame, after removing the dissipative term $\mathbf{f_{vi}}$ added to the control input $\mathbf{u_i}$ of each robot.

- **Test 5** Add a random vector to the control force of each robot to simulate a swarm of insects, animating each robot with Brownian Motion.

- **Test 6** Simulate and represent a Hierarchical Organization with 4 separate groups of robots.

- **Test 7** Simulate a formation with non-holonomic robots.

4.2.1 Experimenting with Interaction Potentials

Test 1 - Simulating the Formation with IPs V_{I1},V_{I2} and V_{I3}

The formation was simulated with several holonomic unit mass robots each having a 0.25m radius, using the Vehicle-to-Vehicle IPs V_{I1},V_{I2} and V_{I3}. These potentials are evaluated based on how the formation responds to the control commands, how it converges to a static configuration and how it performs in avoiding inter-vehicle collisions. Table 4.1 contains the simulation parameters that parameterize the IPs. The dissipative force, given in equation (2.11), will be $\kappa = 2$ multiplied by the velocity of the vehicle. The parameters were hand tuned for a stable formation keeping the robots in the arena, and maintaining a "comfortable" time response to alterations of the equilibrium state, which was evaluated thought the user control interface.

Test 2 - Different Configuration Parameters for Vehicle-Leader IPs

Different configuration parameters are used for the vehicle-vehicle and vehicle-leader IPs, to check how the formation reacts when the distance of influence d_1 of the vehicle-vehicle IP is reduced, compared to the distance of influence of the vehicle-leader. The number of vehicles is $N = 25$, including the Leader, and $\kappa = 2$.

IP	ξ	$d_0(m)$	$d_1(m)$
V_{I1}	0.4	3	100
V_{I2}	1.8	3	100
V_{I3}	0.4	3	100

a) Robot-Leader IP

IP	ξ	$d_0(m)$	$d_1(m)$
V_{I1}	0.06	3	100
V_{I2}	0.8	3	100
V_{I3}	0.01	3	100

b) Robot-Robot IP

Table 4.1: Simulation Parameters for IPs V_{I1},V_{I2} and V_{I3}.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	10	4.0	100.0
vehicle-vehicle	0.5	2.0	3.0

Table 4.2: Test 2 - Configuration Parameters. $d_{1(\text{vehicle-vehicle})} \ll d_{1(\text{vehicle-leader})}$.

Test 3 - Tracing the Robots

In this test we add a trace to each robot to show how the formation converged to its equilibrium configuration, and how the robots follow a leader in translation.The number of vehicles is $N = 13$, including the Leader, and $\kappa = 2$.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	5.0	3.0	100.0
vehicle-vehicle	0.5	3.0	100.0

Table 4.3: Test 3 - Configuration Parameters.

Test 4 - The 3-body Gravitational Problem

This test is presented as a mere curiosity. It is based on the similarity that a formation composed by 3 elements interacting with null dissipation, *i.e.*, $\kappa = 0$, has to the 3-body classical problem in celestial mechanics, a problem that has been studied by Isaac Newton and leading mathematicians over the last two centuries. It focuses on how three celestial bodies move under their mutual gravitational and represents an example of chaos in nature. The test aims on verifying the tracks taken by the robots in a finite time frame.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.1	5.0	100.0
vehicle-vehicle	0.01	3.0	100.0

Table 4.4: Test 4 - Configuration Parameters.

Test 5 - Using Random Forces

When observing a swarm of flying insects it seems that the individuals are animated with some kind of random walk while trying to avoid collisions between each other. This behavior is imitated in this test. Each robot will be animated by adding a random vector $\tau = [x_{rand}, y_{rand}]$ to the control input given by equation (2.11). This functionality can become useful in some applications where the formation should cover a certain unknown area in missions like, *e.g.*, scientific mapping, search and rescue and gradient tracking. The idea is to simulate the "cloud" of insects with the formation and deploy it to investigate and explore a target area. The random variables x_{rand}, y_{rand} present a uniform distribution in the interval defined the configuration parameters MinRandForce and MaxRandForce. For this example, we used the interval $[-3, 3]$ N. The number of vehicles is $N = 23$, including the Leader, and $\kappa = 2$.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	5.0	3.0	100.0
vehicle-vehicle	1.0	5.0	100.0

Table 4.5: Test 5 - Configuration Parameters.

Test 6 - Hierarchical Organization

In this test we implement a hierarchical formation with 2 levels. The formation is composed by a total number of 29 vehicles, that are organized into 4 separate groups $C_1, \ldots, C_4$, operating in $level_1$. Robots included in each group will be associated by color in the "`World.bmp`" file, where each robot is represented by a Red, Green, Black or Blue pixel. The first robot to be included in each group will be selected as the group leader, and will take part in $level_2$, the second formation level. The first leader to be selected in this level will respond to the operators commands, to guide

the formation. Algorithm 3 summarizes the program, with the alterations done to algorithm 2.

Table 4.6 summarizes the IP configuration parameters used for $level_1$ and $level_2$. Within each level, the same parameters are used for vehicle-vehicle and vehicle-leader IPs.

IP	ξ	$d_0(m)$	$d_1(m)$
$Level_1$	0.5	2.0	100.0
$Level_2$	0.5	12.0	100.0

Table 4.6: Test 6 - Configuration Parameters.

4.2.2 Using Non-Holonomic Robots

In this simulation the ideas are extended to a team composed by differential drive robots, and the IF is applied to a HP placed at the robots front boundary. Figure 4.2 shows the representation of the differential drive robot drawn in the simulation.

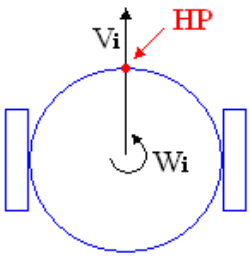


Figure 4.2: Differential drive robot simulation representation.

Test 7 - Simulating the Formation

The static and dynamic behavior of the formation is analyzed with a dissipative force magnitude constant of k = 2. Table 4.7 contains the IP simulation parameters used to present the equilibrium configuration of a formation with (20+1) robots. Table 4.8 contains the parameters used in a formation with $(6 + 1)$ robots. This last example will be used to show how formations composed by non-holonomic robots move in space.

Algorithm 3: Matlab Simulation Program

```
Initialize Simulation Parameters
Read World.bmp and Setup World Configuration
Setup Remote Communication Interface
Wait for Incoming Connection
Start Timer
while TerminationCondition = false do
    Read and Decode Control Messages from Remote Console
    Process Control Message:
    begin
        Verify Termination Condition
        Calculate Leader Velocities
    end
    Calculate Elapsed time δ_t
    Update Leader[1] State Variables using δ_t
    Process Level 2:
    begin
        for each robot in level_2 (level_1 leader) do
            for all other robots in level_2 do
                Update IF with the Force from the Interaction with this robot
            Add and additional dissipative force f_vi to IF
            Update the State Variables of the robot using IF and δ_t
    end
    Process Level1:
    begin
        for each group C_1 to C_4 do
            for each robot in the Group do
                Calculate IF (see algorithm 2)
                Update the State Variables
    end
    Draw World
    Draw Robots in level_2 (leaders of level_1)
    Draw Level1:
    begin
        for each group C_1 to C_4 do
            Draw the team
    end
Close Connection with Remote Interface
```

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	20.0	5.0	100.0
vehicle-vehicle	5.0	4.0	100.0

Table 4.7: Test 7a - Configuration Parameters for a formation of (20+1) robots.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.8	4.0	100.0
vehicle-vehicle	0.8	3.0	100.0

Table 4.8: Test 7b - Configuration Parameters for a formation of (6+1) robots.

4.3 Obstacle Avoidance

To explore the ideas presented in Chapter 3, obstacles are now included in the simulation. Every black pixel in "`world.bmp`" file is considered an obstacle and so, by editing this file, the existing obstacles may be modified or new obstacles can be added to the scenario. Figure 4.3 shows an example of a world containing 3 obstacles with different geometries.

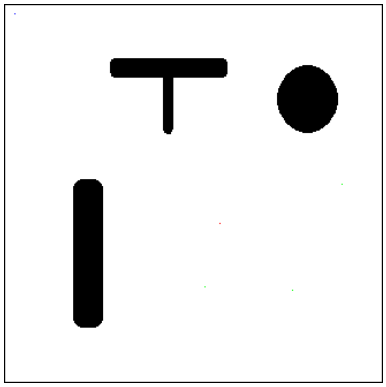


Figure 4.3: "`World.bmp`" file, with obstacle representation.

To simulate the ODL described in Section 3.1, each robot will include a certain number of "detectors" placed on its boundary. A configurable simulation parameter contains the vector with the radial positions of the detectors, declared as $sPlc = [\frac{-3\pi}{4}, \frac{-\pi}{2}, \frac{-\pi}{4}, 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi]$. Each detector will detect an obstacle at a distance within a range defined by $sRange = 4(meters)$, and the readings, associated with the detectors, are retrieved in a vector after a call to a function named `getSonarData`(...).

These values are obtained based on the world and on the robots current configuration. The detection process can be represented graphically as shown in figure 4.4.

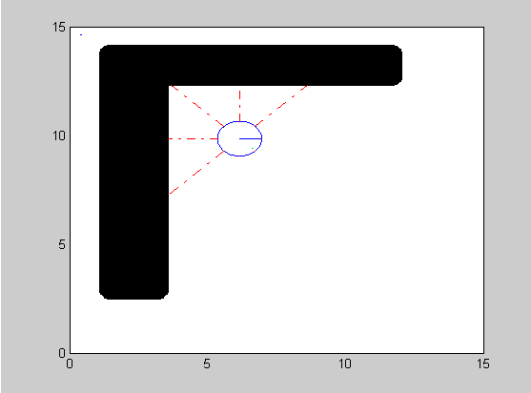


Figure 4.4: Sonar detection simulation.

The inter-vehicle desired distance is set to a value higher than the robots sensor ring range so that this way, any value measured by the robots sensors will be due to an obstacle. The distance to the obstacles will be used to calculate the respective repulsive forces acting on each robot, as previously described. A repulsive force will also be used as a control input to the leader, combined with the velocity input imposed by the remote operator, to facilitate the process of guiding the leader throughout the obstacle field without hitting the obstacles. Table 4.9 contains the default repulsive APF parameters used in the simulations. Parameter η is the scaling factor and d_l is the distance of influence of the repulsive potential.

	η	d_0 (m)
Leader	100	4
Other Robots	40	2

Table 4.9: Repulsive APF Parameters.

Brief description of each test:

- **Test 8** Aims to verify the behavior of the formation, when repulsive potentials are included for obstacle avoidance.
- **Test 9** Test 8 is repeated, using non-holonomic robots.

- **Test 10** In this test robots will track the leader (Version I of the tracking algorithm), to avoid getting stuck in local minima.

- **Test 11** Same as test 10, with some modifications to the tracking algorithm.

- **Test 12** Test 11 is repeated, using non-holonomic robots.

4.3.1 Using Repulsive Potentials

Test 8 - Testing Potentials for Obstacle Avoidance

The potentials for obstacle avoidance, described in Chapter 3, are simulated and evaluated using formations with 1 and 5 robots, using $\kappa = 2$. The maximum control force and velocity are also limited, for all robots, to a maximum value of $fMax = 40$N, and $vMax = 6ms^{-1}$.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.05	6.0	100.0
vehicle-vehicle	0.05	6.0	100.0

Table 4.10: Test 8 - Configuration Parameters.

Test 9 - Testing Obstacle Avoidance, with Non-Holonomic Robots

The parameters of Test-8 are now applied to a formation with non-holonomic differential drive robots. The size of the arena will be changed to 40m, for better visualization of the robots. Examples are shown for 1, 5 and 10 robots.

4.3.2 Tracking the Leader

Test 10 - Following the Leader using a Track Log (Version I)

The ability to follow the Leader's track by using a track log and starting at the leader's position, is now considered for a formation composed by a single robot and a leader. The simulation parameters are the same as the ones used in Test 8. The configuration parameter for the clearance condition is given by $\xi_{max} = -0.8$.

Test 11 - Following the Track Log (Version II)

Some modifications are made to algorithm 1 to minimize the clearance problems verified in the previous test. In the second version of the leader following code, the starting position in the clearance search process is no longer the leader's current position, but the CAP. The algorithm will keep the Q_{LOG} index of this point to use it in the successive iterations, searching the log upwards, towards the leader's position, until it finds a point in the track that does not verify the clearance condition. Then, the algorithm will roll back towards a point that does. Algorithm 4 summarizes this part of the algorithm. The objective is to try and minimize the oscillations between two opposite points, by forcing the robot to track clearance all the way up to the leader. It is possible two adopt different values ξ_{max1} and ξ_{max2} for whenever the clearance search process is rolling up or down. Normally, $\xi_{max1} < \xi_{max2}$, *i.e.*, the clearance condition is more demanding (larger β aperture) when the algorithm is searching for a point that verifies the condition, than when rolling up towards the leader. The negative aspect of the algorithm is that with this approach, a short-circuit to the path can no longer be assumed. If a robot looses clearance it must inevitably follow the track, even if direct clearance to the leader is made available through another path.

Algorithm 4: Tracking the Leader. Version II

```
Q_LOG(headPosition) = LeaderCurrentPos
posIndex = indexOf(currAttractionPoint)
while posIndex != beginningOf(Q_LOG) do
    if clearanceCond(posIndex,ξ_max1) == false then
        break
    posIndex = posIndex - 1
while posIndex != endOf(Q_LOG) do
    if clearanceCond(posIndex,ξ_max2) == true then
        currAttractionPoint = Q_LOG(posIndex)
        modifySimulationParameters
        break and use currAttractionPoint
    posIndex = posIndex + 1
```

The following parameters were used: $fMax = 40$N, and $vMax = 8ms^{-1}$,$\xi_{max1} =$

-0.8 and $\xi_{max2} = -0.4$.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.1	6.0	100.0
vehicle-vehicle	0.4	6.0	100.0

Table 4.11: Test 11 - Configuration Parameters.

Test 12 - Following the Track Log, with Non-Holonomic Robots

Test 11 is repeated, using 3 non-holonomic robots and 1 leader.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.08	6.0	100.0
vehicle-vehicle	0.08	6.0	100.0

Table 4.12: Test 12 - Configuration Parameters.

5 Results

In this chapter, the results obtained for the conditions set up in the previous chapter are presented. Most of the results are shown in the form of snapshot images taken from Matlab's graphical output. Conclusions are presented for each result.

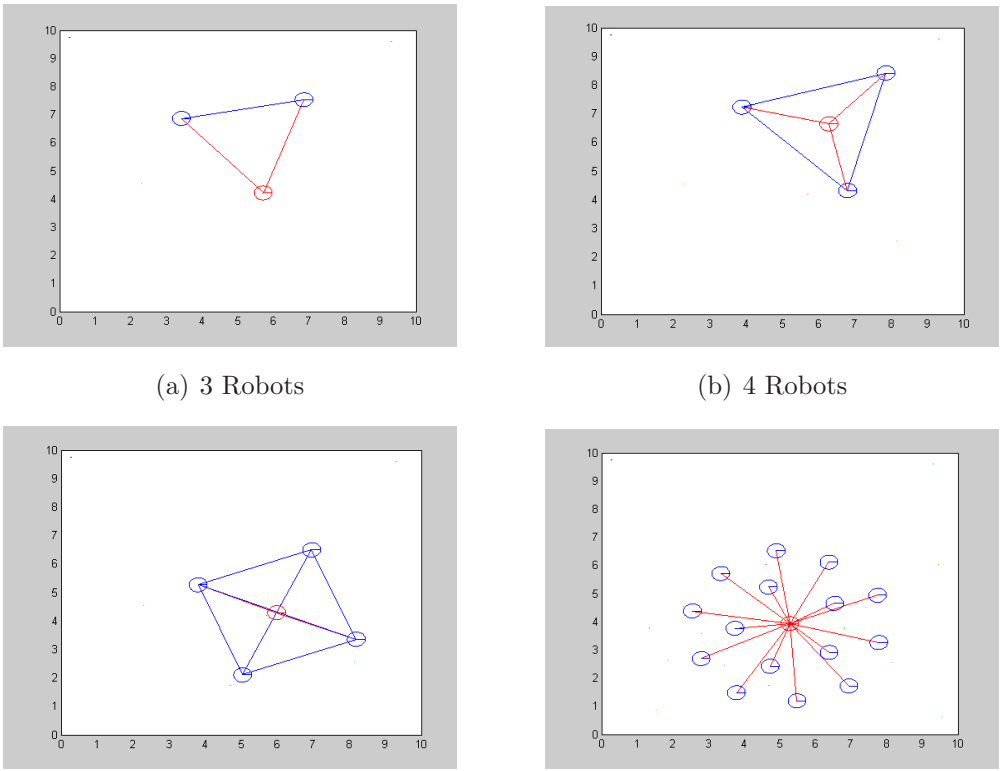
5.1 Formation Framework Experiments

5.1.1 Experimenting Interaction Potentials

Test 1 - Simulating the Formation with IPs V_{I1},V_{I2} and V_{I3}

In this test the Vehicle-to-Vehicle IPs V_{I1},V_{I2} and V_{I3} were evaluated. Compared to the present simulation setup a practical approach, or a more detailed simulation, may reveal other details and metrics to evaluate and compare the full utility of each IP. The described results lack this metric and are purely based on the taken observations. It was verified that V_{I1} and V_{I3} both converge to the same static equilibrium configuration and present a very similar dynamic response. The sensation given to the operator is like moving a set of elements interconnected by elastic springs. With V_{I1} when the leader assumed a certain velocity towards another robot the IF between

them, in some situations, was not sufficient to prevent a collision. V_{I3} reinforces inter-vehicle collision avoidance and for this purpose, compared to the other IPs, it obtained a superior performance. The evaluation of V_{I2} confirmed that the formation does not converge asymptotically to an equilibrium configuration and that the robots oscillate around their equilibrium point do to the discontinuity of the resulting force. As to the dynamic response all robots translate with the same constant velocity (the spring effect is lost) and several inter-vehicle collision situations were verified. Unless stated otherwise, V_{I3} will the selected IP for further formation experiments. Figure 5.1 shows different sized formations in a static equilibrium configuration, using V_{I3}. The IF links are represented by lines connecting the center position of the vehicles, with blue lines representing vehicle-vehicle IF links and red lines, vehicle-leader links.



(a) 3 Robots

(b) 4 Robots

(c) 5 Robots

(d) 16 Robots

Figure 5.1: Formation stable equilibrium configurations showing vehicle-vehicle IP connections(blue lines) and vehicle-leader(red lines).

A final observation is related to the magnitude of attraction of the whole formation to the equilibrium configuration that increases with the number of robots. The number of force components acting on a formation of $N > 1$ robots is given by $N(N-1)$

and to correct this dependency, the potential scaling constant ξ can be multiplied by a factor of $\frac{1}{N(N-1)}$. Figure 5.2 shows the stable equilibrium configuration of a larger formation.

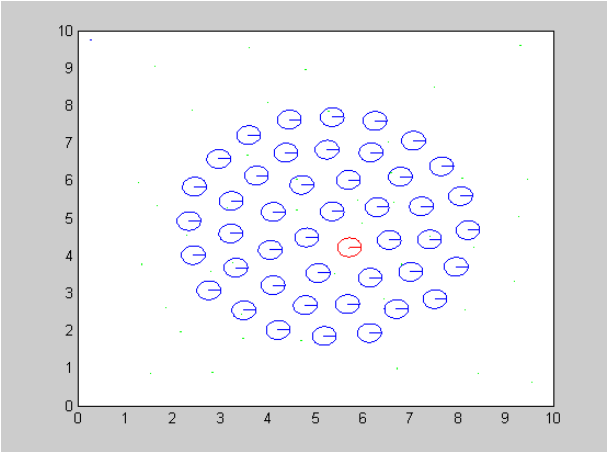


Figure 5.2: Formation stable equilibrium configuration with 45 robots.

Test 2 - Different Configuration Parameters for Vehicle-Leader IPs

Figure 5.3 shows the stable formation configuration after modifying the IP configuration parameters d_0 and d_1. Each robot is no longer connected to all the other teammates, but only to the limited set of vehicles that falls inside its range defined by the new value for d_1, with $d_{1(\text{vehicle-vehicle})} \ll d_{1(\text{vehicle-leader})}$.

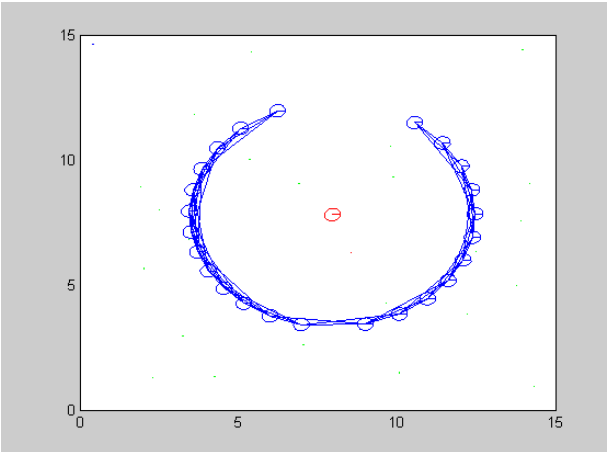
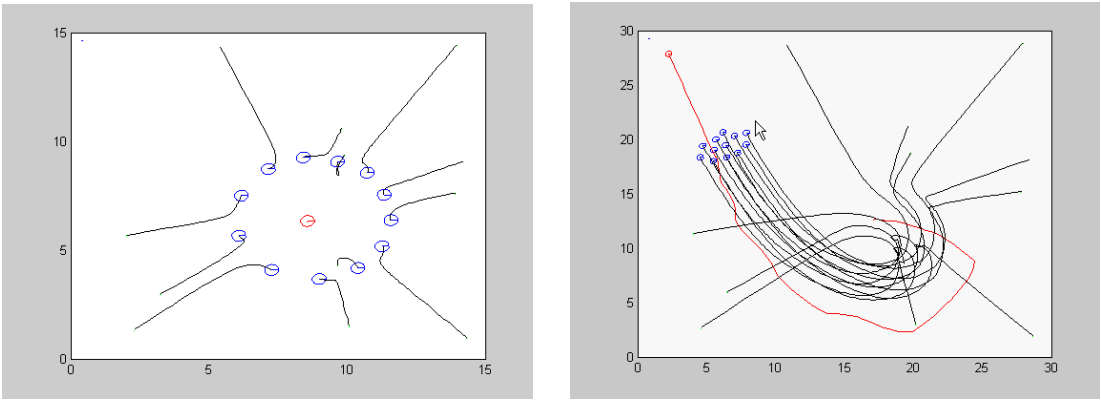


Figure 5.3: Formation stable equilibrium configuration using different d_0 and d_1 parameters for the vehicle-vehicle and vehicle-leader IPs.

Influenced only by near neighbors, robots will tend to get aligned in the circle around the leader, defined by the specified distance of equilibrium d_0 of the vehicle-leader IP.

Test 3 - Tracing the Robots

The traces of each robot in the formation are depicted in figure 5.4. They show the robots trying to converge to a equilibrium configuration in a situation where the leader holds a static position, and when the leader is moving and "dragging" the formation.



(a) Converging to a static equilibrium configuration

(b) Following the leader

Figure 5.4: Tracing the dynamic behavior of the formation.

Test 4 - The 3-body Gravitational Problem

A formation interacting with null dissipation will not converge asymptotically to a stable configuration. A example of this situation is presented in Figure 5.5 that shows a formation with two robots, their corresponding trajectories and a static leader. The interaction between these robots exhibits all of the hallmarks of chaos. In particular, the outcome of any given interaction depends sensitively on the initial conditions, a characteristic of all chaotic systems.

Test 5 - Using Random Forces

Animating the robots with Brownian motion can be a solution to covering a certain target area with the formation when the mission includes exploring. The solution

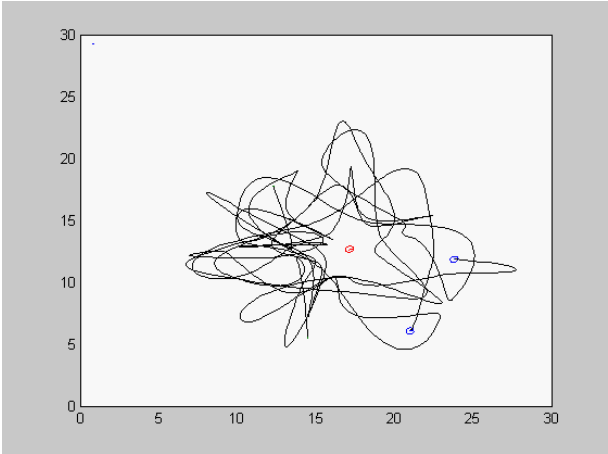


Figure 5.5: 3-Body problem. Formation interacting with null dissipation ($\kappa = 0$).

maintains the fundamental motivation of using IPs to control the formation, by the way that this solution translates to the control laws necessary to implement this behavior. Figure 5.6 shows the tracks of the robots covering the area around the leader, taken after a certain time interval.

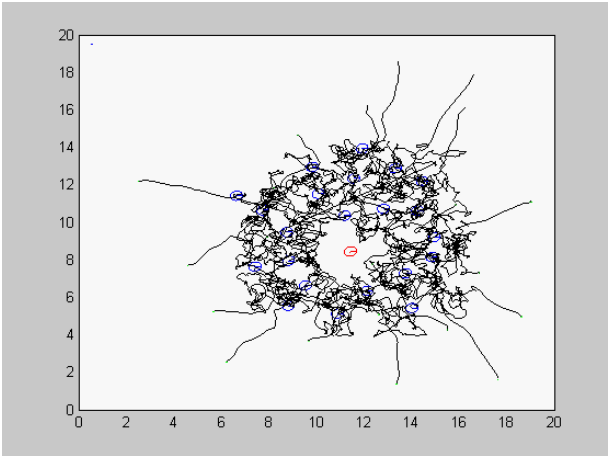


Figure 5.6: Tracks of a formation covering and exploring an area with Brownian motion (N° robots = 23).

Test 6 - Hierarchical Organization

A hierarchical formation with 2 levels was tested and figure 5.7 shows the formation with the 4 different groups in an equilibrium configuration. The IF links at $level_1$ are represented with green lines and red lines represent the links at $level_2$. It was

verified that parameters for both levels should be chosen with some care to prevent inter-vehicle collisions between vehicles belonging to different groups. This becomes even harder to prevent, if the formation is in translation. It was verified that, in many situations, the "elastic behavior" of the IF links would get some groups critically close together. A "rigid" configuration for the links at $level_2$ can be used to keep the groups at a certain safe distance and orientation while moving the formation.

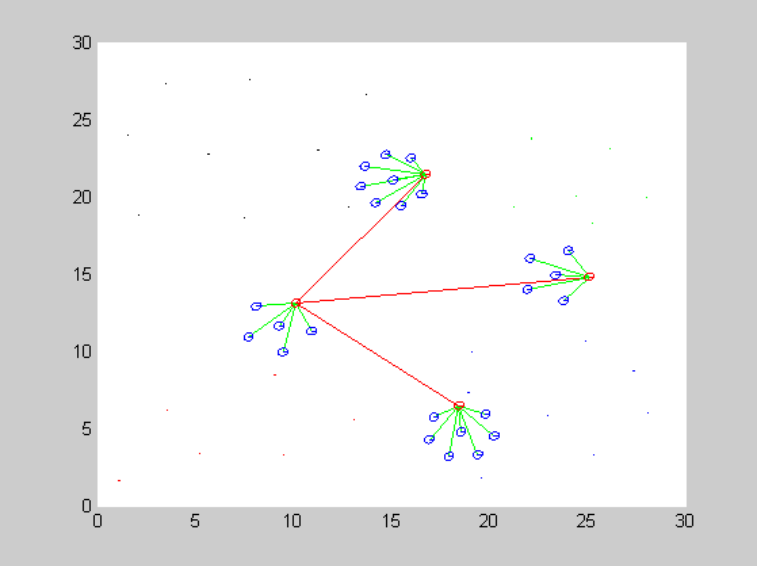


Figure 5.7: 2-Level Hierarchical Organization, with 4 groups $C_1, \ldots, C_4$

5.1.2 Using Non-Holonomic Robots

Test 7 - Simulating the Formation

With simulation, differential drive formations obtain results similar to the ones obtained with holonomic formations. It was verified that the robots orientation is uncontrollable and tends to get aligned, while moving, with the followed direction. This is not considered a limitation when the objective is to deploy the formation to a certain target area. Figure 5.8 shows a stable equilibrium configuration of a formation with 20 robots. Figure 5.9 shows the trace of each robot.

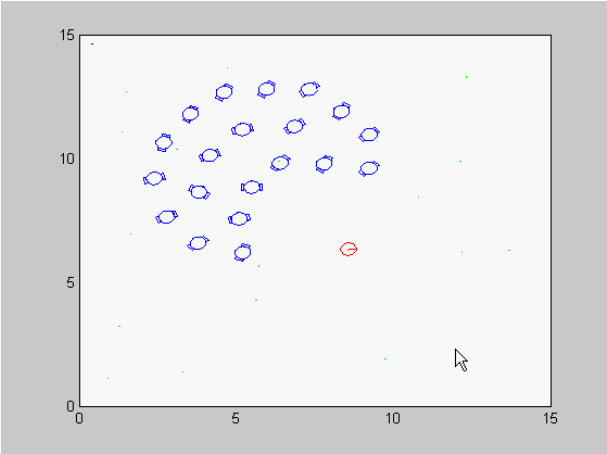


Figure 5.8: Stable equilibrium configuration of a team with differential drive robots.

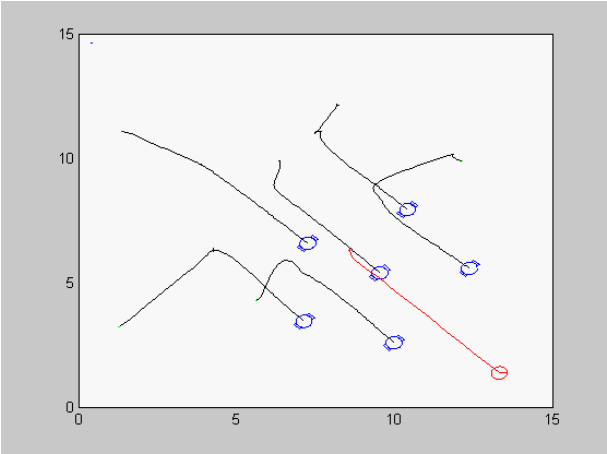


Figure 5.9: Tracing a non-holonomic formation.

5.2 Obstacle Avoidance

5.2.1 Using Repulsive Potentials

Test 8 - Testing potentials for obstacle avoidance

The control force acting upon each robot now includes a repulsive component, associated with the obstacles sensed within a limited range around the robots perimeter. When the leader moves, the vehicles will try to follow complying, while moving, with the obstacles that they encounter in their path. Figure 5.10 shows examples of several paths, followed by a single robot. The yellow line indicates a IF link to the reference point that is currently being used by the vehicle to follow the leader. The robot will

try to keep up to this task, even if the leader is on the other side of an obstacle, and in many situations, the robot will short-circuit the leader's path to keep in formation. This is by no means a limitation and is, for the proposed objectives, a desired behavior.

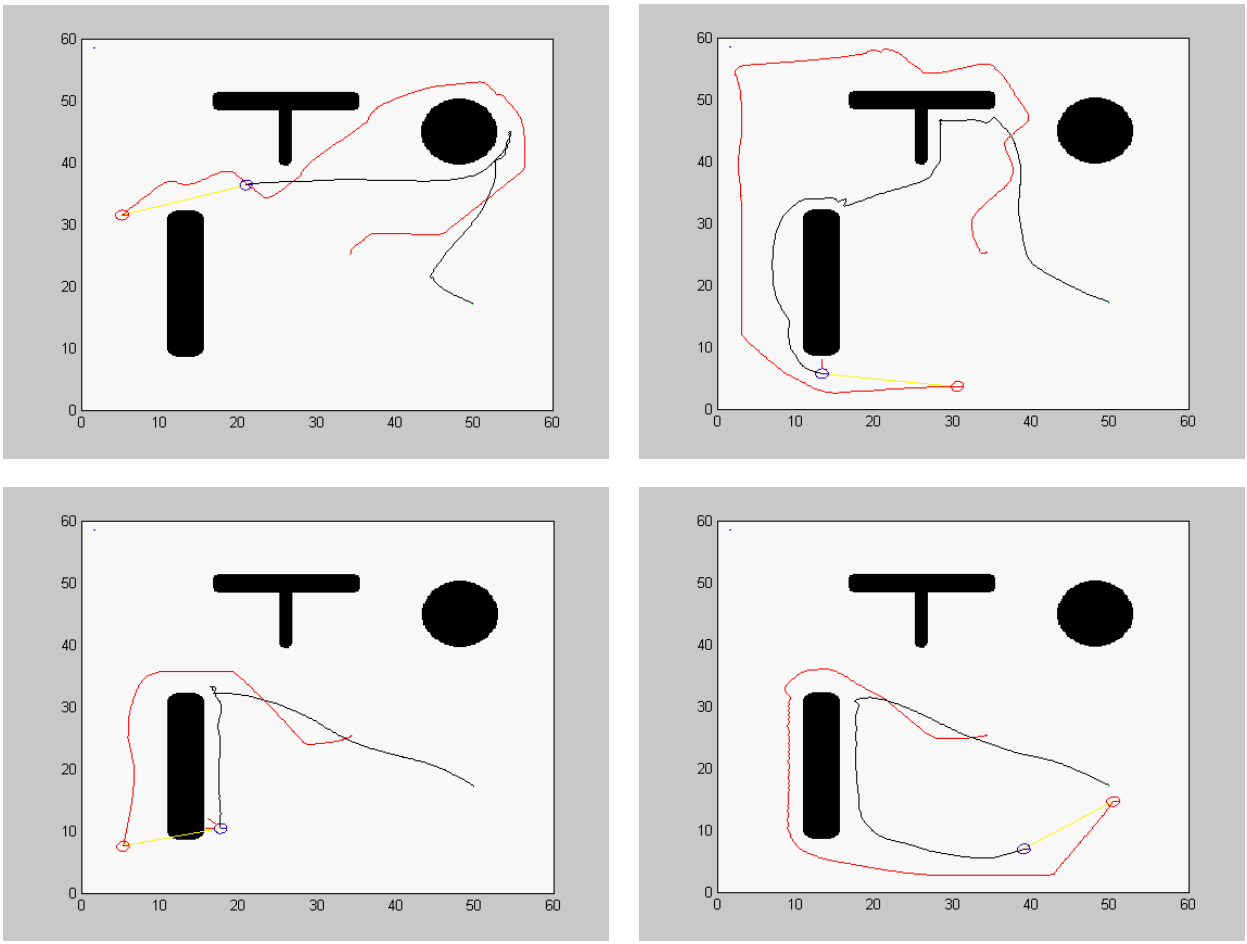
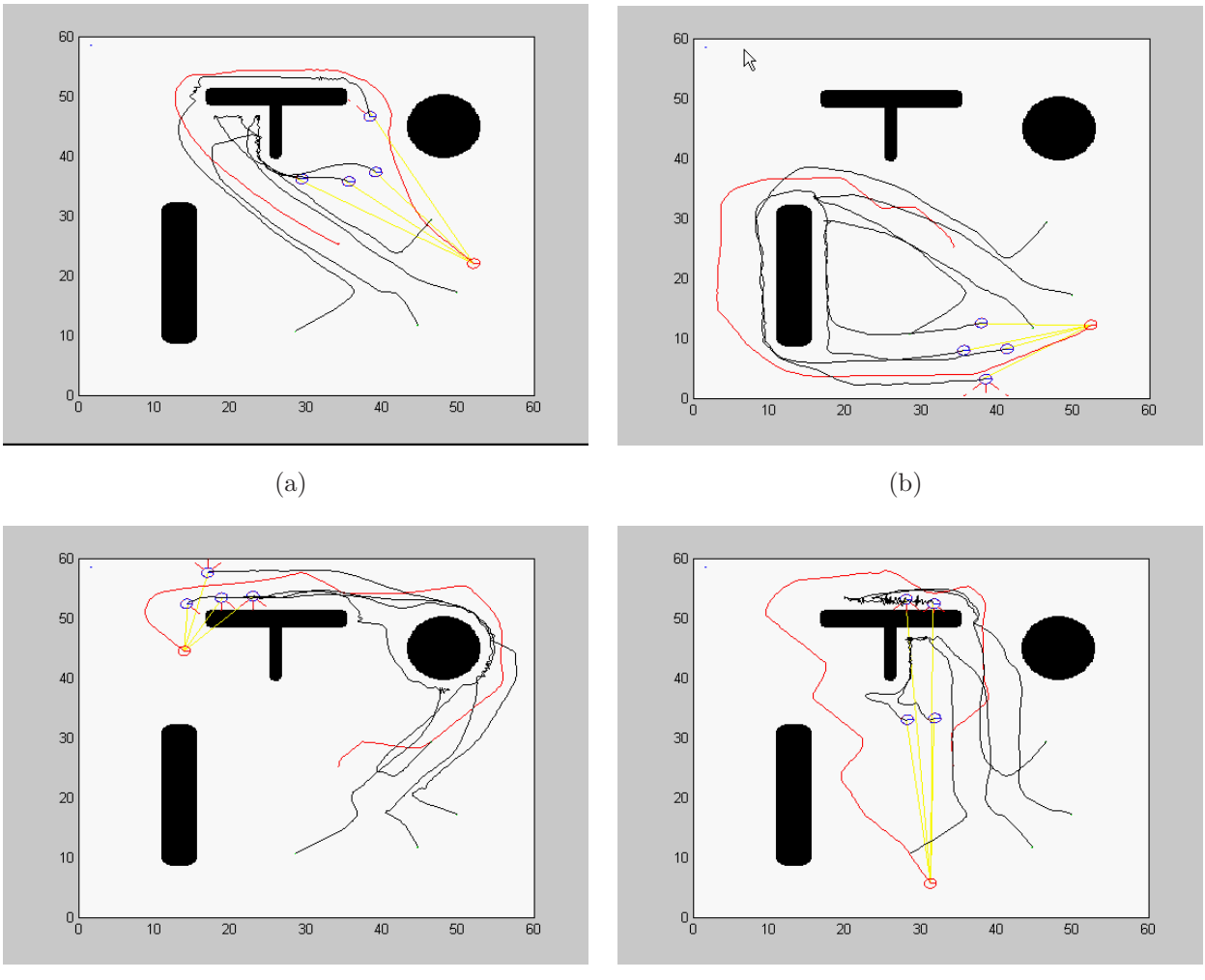


Figure 5.10: Paths followed by a formation with a leader and a follower, with obstacle avoidance.

Figure 5.11 shows the path followed by 5 robots (4 robots and 1 leader) in a obstacle field. In some cases the presence of obstacles forces the formation to split making vehicles follow different directions around the obstacles. In other situations, vehicles can even get stuck behind. Figure 5.11(d) shows an interesting situation to be considered. It shows that, although two of the vehicles have direct clearance to follow the leader, they are immobilized due to their need to also keep in formation with the other vehicles that got stuck behind the obstacle. The obstacles presented in

the examples have a geometry that will enable, sooner or later, with some probability, the possibility for the vehicles to regain clearance towards the direction of the leader. Until this clearance occurs, they are retained back without any kind of orientation as to how to get out of the local minimum situation. Figure 5.12 shows a situation where the geometry of the obstacle traps the formation. For these vehicles to regain clearance, the leader would have to turn back and pull the formation out of this area. If the leader follows ahead, the robots will no longer be able to leave this trapping situation.



(a) (b)

(c) (d)

Figure 5.11: Paths followed by 5 robots, with obstacle avoidance.

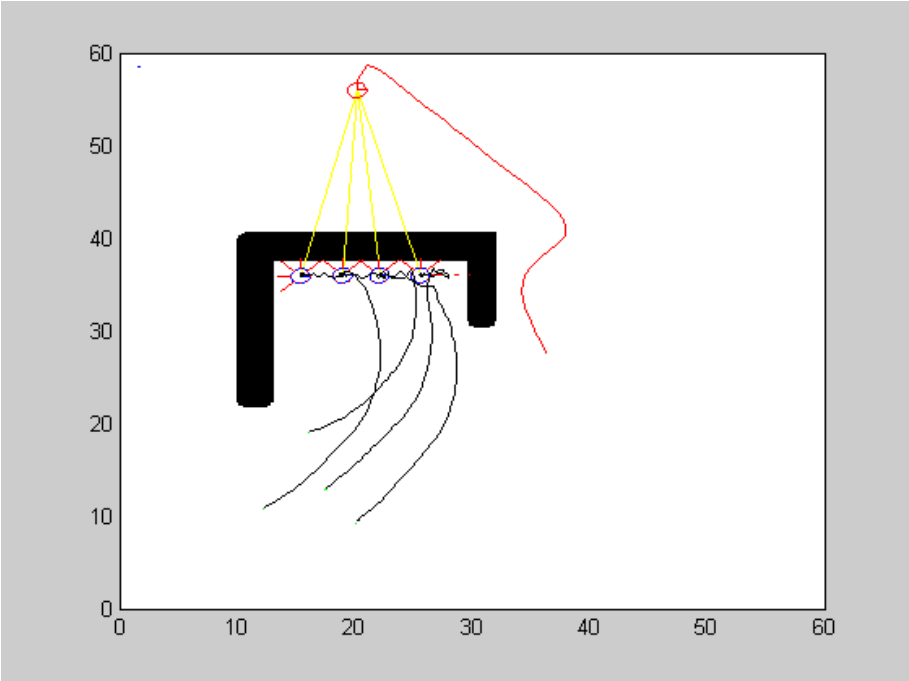
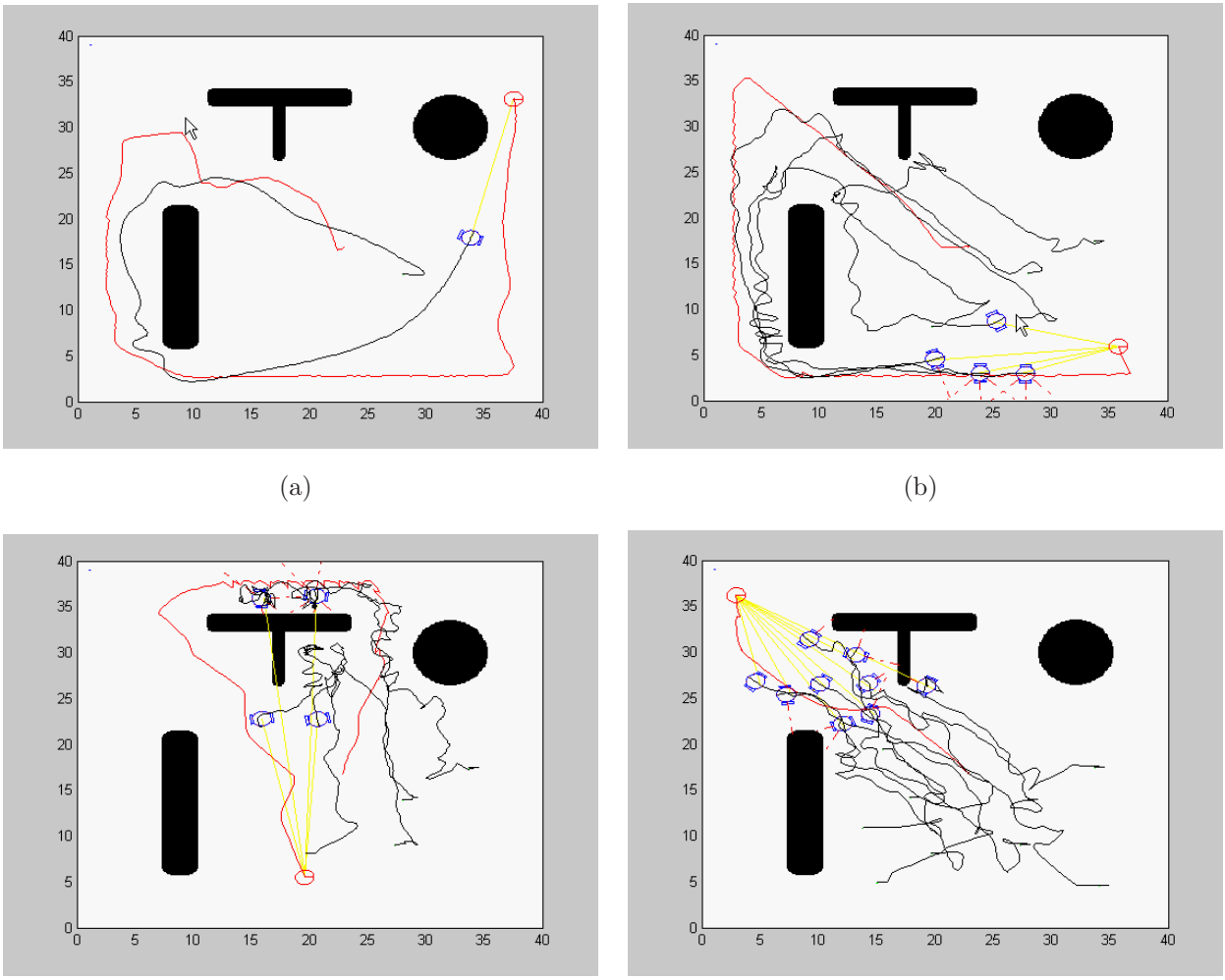


Figure 5.12: Formation gets trapped behind an Obstacle.

Test 9 - Testing Obstacle Avoidance, with Non-Holonomic Robots

Teste-8 is repeated using differential drive robots. Figure 5.13 shows the tracks of 4 formations, that are drawn from the positions assumed by the HP of each robot. The tracks show some oscillations that are not verified with holonomic vehicles. The intensity of the oscillations outstands when the formation converges to its equilibrium configuration, and when robots are subjected to obstacle repulsive forces. Figure 5.14 shows a situation where a robot is oscillating due to the interaction between the FF and repulsive IPs. With the IF applied to the off-axis HP, non-holonomic robots do not seem to stabilize like the other robots, where the application point is at the center of the vehicle. The oscillations were minimized with some hand adjustments to the simulation parameters. The maximum velocity limit is reduced to $vMax = 2ms^{-1}$, and the dissipative force magnitude is increased to $\kappa = 5$. Table 5.1 shows the modified IP parameters. Figure 5.15 presents two examples with the tracks of the robots after these adjustments.



(a) (b)

(c) (d)

Figure 5.13: Paths followed by non-holonomic robots, with obstacle avoidance.

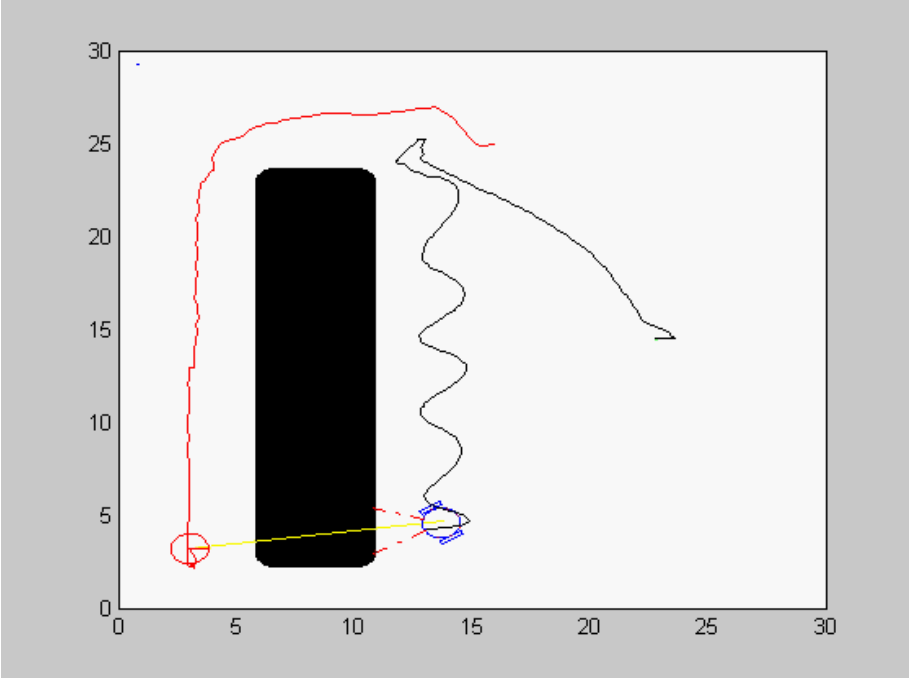
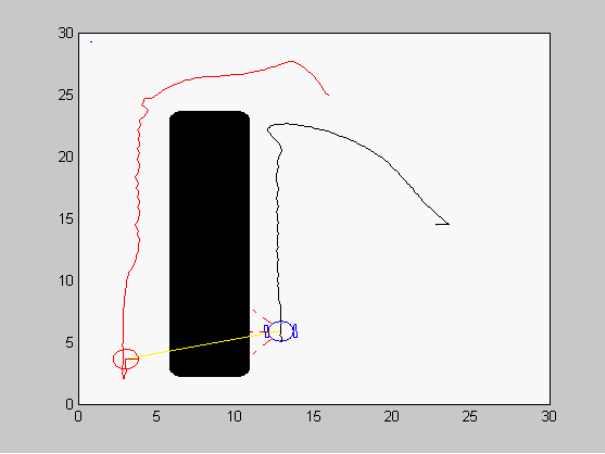
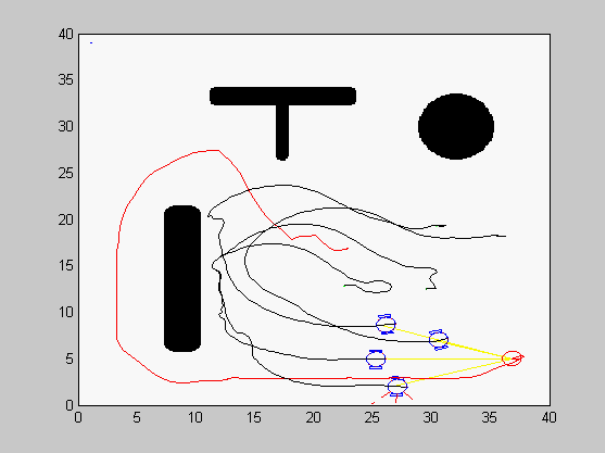


Figure 5.14: Oscillating paths with non-holonomic robots.



(a)



(b)

Figure 5.15: Paths after parameter adjustments for Oscillation attenuation.

IP	ξ	$d_0(m)$	$d_1(m)$
vehicle-leader	0.9	6.0	100.0
vehicle-vehicle	0.5	5.0	100.0

Table 5.1: Test 9 - New Configuration Parameters.

5.2.2 Tracking the Leader

Test 10 - Following the Leader using a Track Log (Version I)

After detecting that an obstacle is obstructing the direct path towards the leader a robot will search for clearance by tracking the leader's position history. Figure 5.16 shows some examples of single robot trying to track the leader after falling back behind an obstacle. The blue lines represent points that fail to verify the clearance condition. The last line is a yellow line and corresponds to the first point to verify this condition. This will be the point considered for the formation in the next iteration step. The algorithm maintains the ability to short-circuit the path around the obstacle if direct clearance to the leader is regained though another path. This is possible because the search algorithm starts from the leader's current position.

It was verified, however, that the clearance validation condition does not always work. In several situations, a valid condition was verified where it shouldn't have been. This happens when the leader translates away following a side-way direction on the opposite side on an obstacle, in such a way that the cone stops containing F_r giving a false idea of clearance. The condition fails and traps the robot behind the obstacle, inducing it into a oscillation mode between two distinctive valid conditions. Figures 5.16(c) and 5.16(d) show examples of this situation, and figure 5.17 shows why it happens, with $\mathbf{F}_{rep}(\mathbf{q_i})$ falling outside the detection cone. Further work must be done to make the clearance condition widely general.

Test 11 - Following the Track Log (Version II)

The modified version of the leader tracking algorithm was tested. Figure 5.18 shows 4 examples with the tracks followed by each robot. It can be verified that the search process can focus on some segment that does not always start at the beginning of the track. Figure 5.18(b) shows a example of the robot finding its way out of the trapping area. Figure 5.19 shows a snapshot sequence of a formation going around an obstacle. The blue lines represent active links and red lines represent broken links.

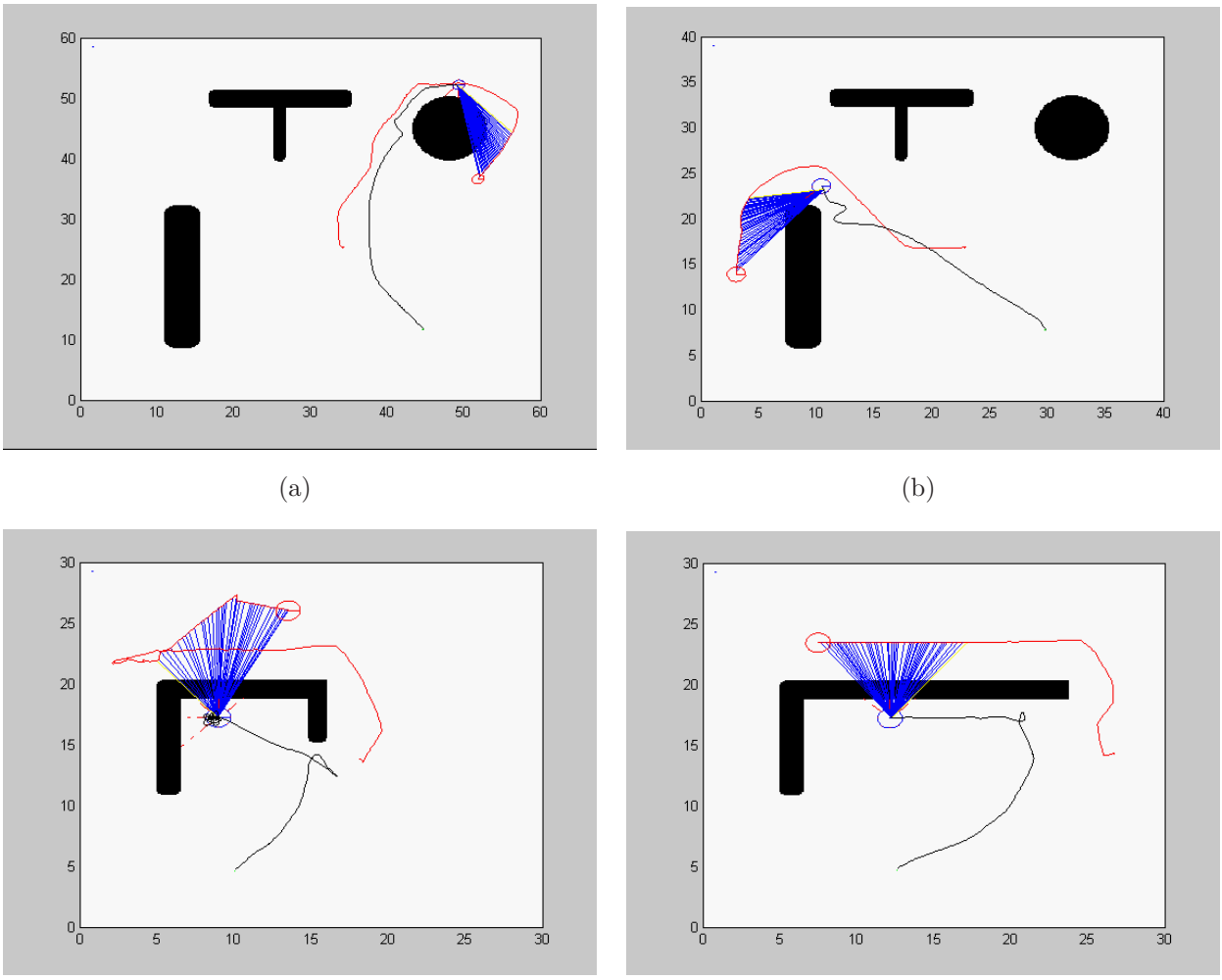


Figure 5.16: 1 Robot following the leader's path (version I)

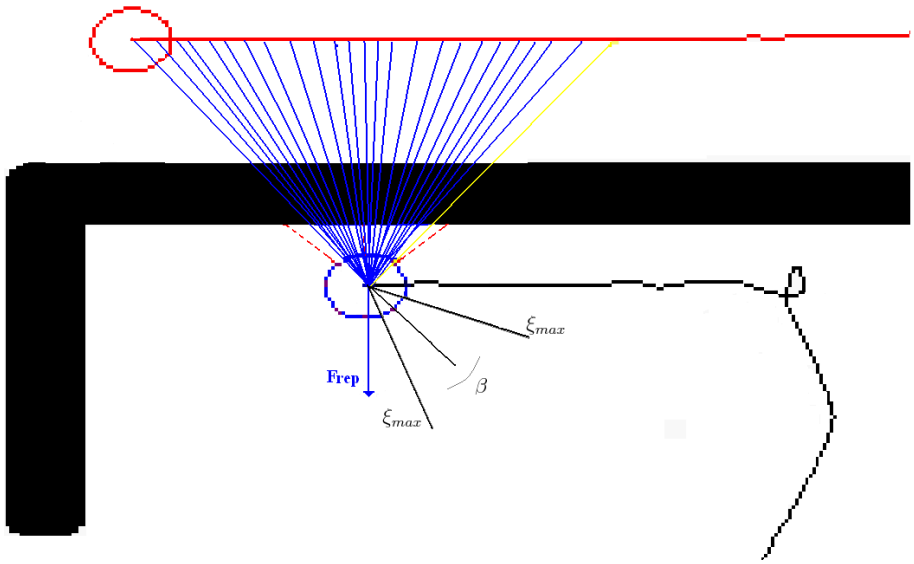
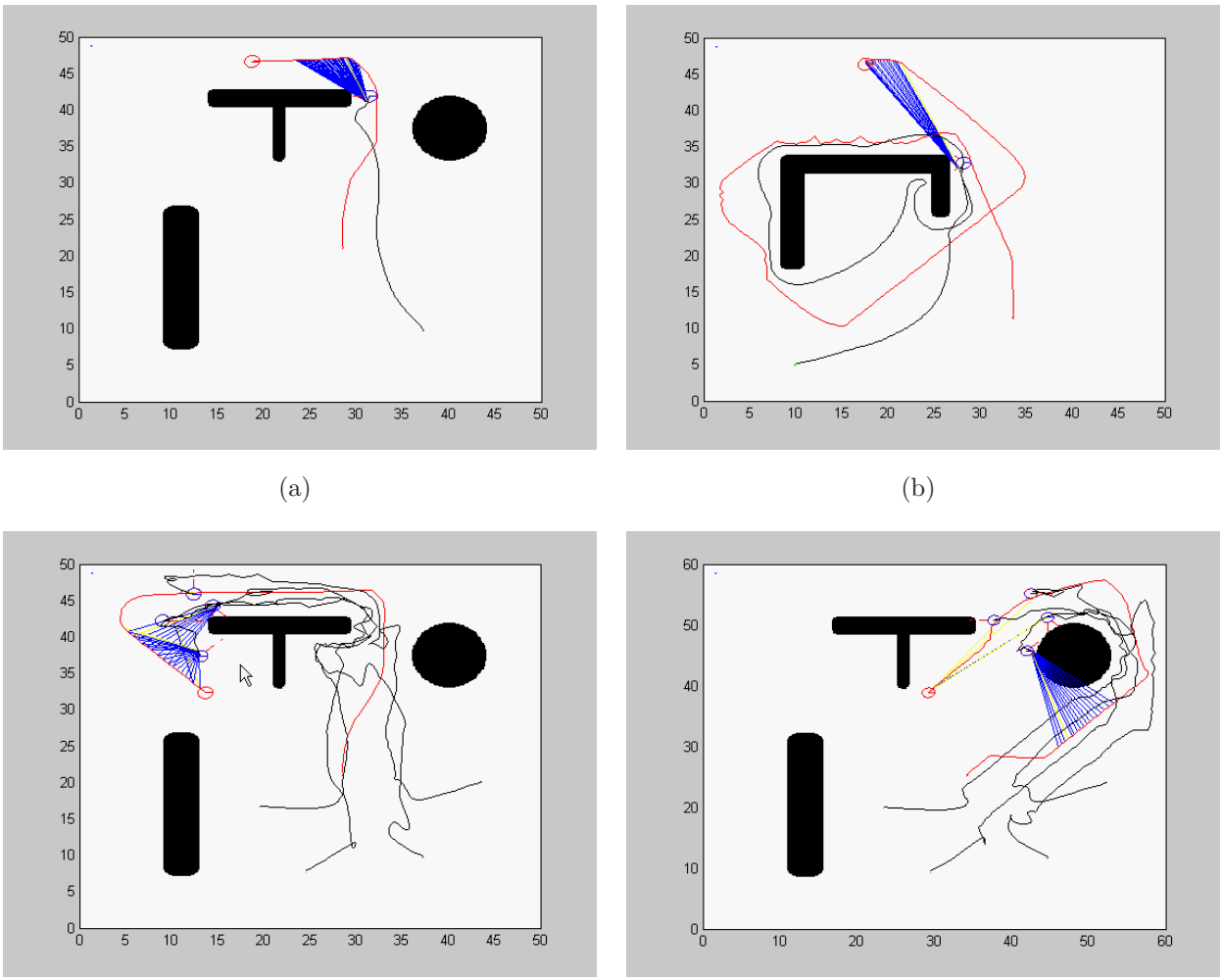


Figure 5.17: The clearance condition fails.

With the presented solution the problem encountered in version I of the algorithm was minimized. But in some sporadic situations, trapping situations still exist. It was verified that this is due to the fact that the tracking process will initiate and operate only when a repulsive vector $\mathbf{F}_{rep}(\mathbf{q_i})$ becomes present. A cyclic situation may occur due to the following sequence of events:

1. A robot gets trapped;
2. robot will search the path for a point with clearance (the CAP);
3. the CAP "pulls" the robot away from the obstacle;
4. the repulsive vector vanishes;
5. robot gets direct clearance towards the leader;
6. direct clearance draws the robot back into the trapping area.

Although this situation does not happen very frequently, some work is still needed on the version II of the algorithm to avoid leaving robots behind, while the formation is moving throughout the obstacle field.



(a) (b)

(c) (d)

Figure 5.18: Robot(s) following the leader's path (version II)

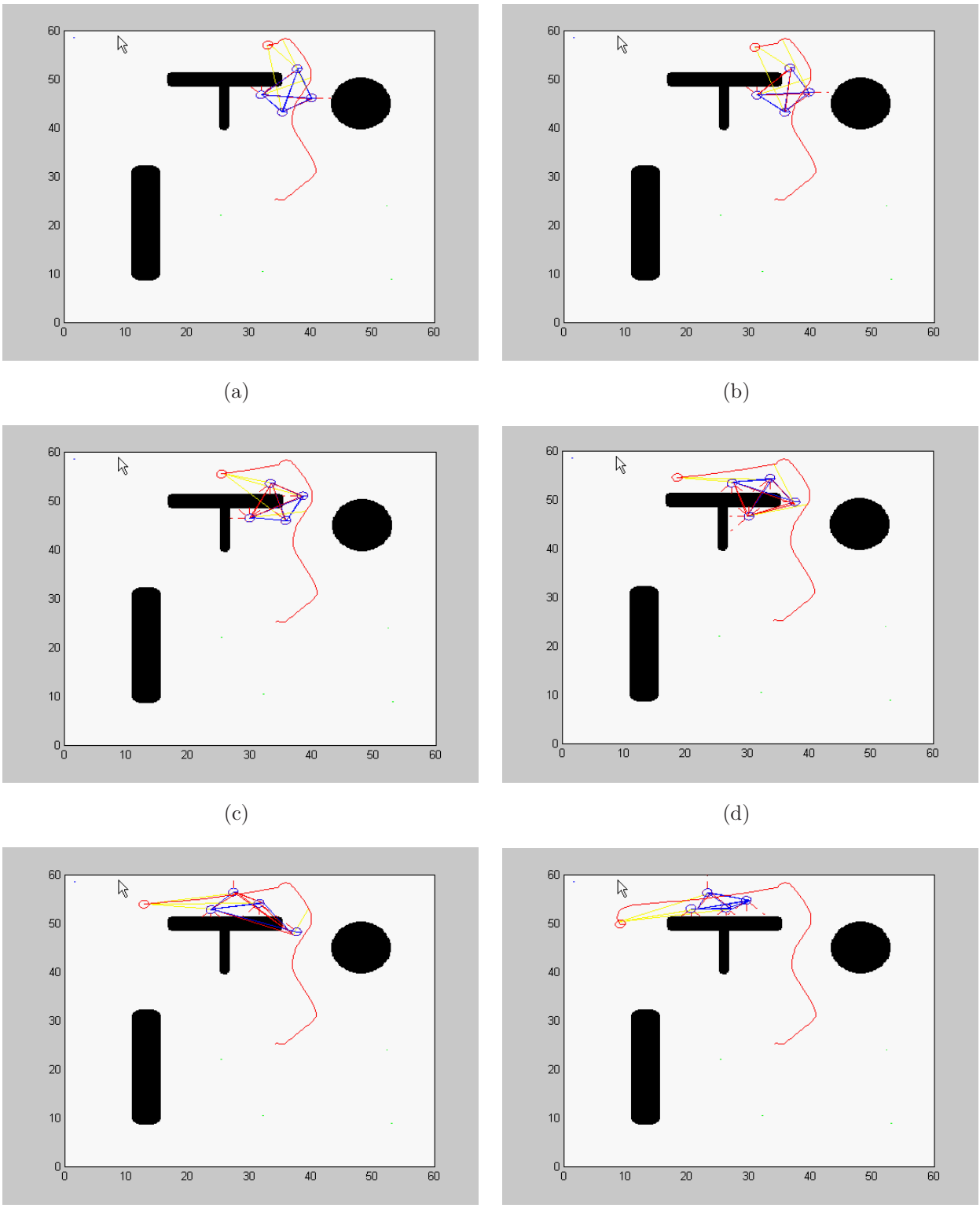
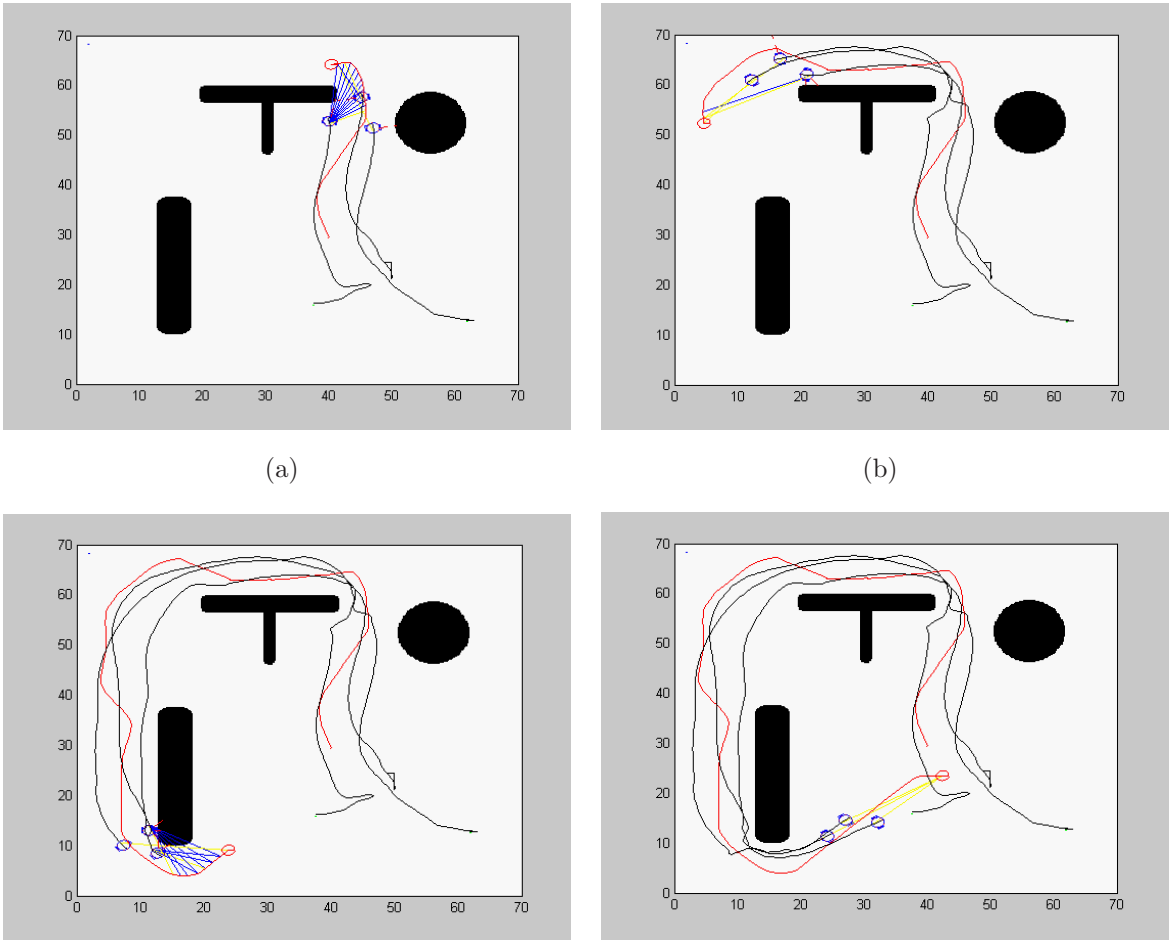


Figure 5.19: Snapshot sequence of the formation following the leader.

Test 12 - Following the Track Log, with Non-Holonomic Robots

Figure 5.20 a snapshot sequence of a formation composed by 3 non-holonomic robots and a leader. The conclusions are the same as the ones presented in the previous test.



(a) (b)

(c) (d)

Figure 5.20: Snapshot sequence of a formation following the leader, with 3 Non-Holonomic Robots.

6

Conclusions and Future Work

This thesis introduced an implementation of a formation control methodology for a team of holonomic and/or non-holonomic mobile robots. A formation is maintained by using artificial potentials for the intra-team distances. Each robot has full knowledge of the configuration of the whole formation, and travels behind a leader through a field scattered with unknown obstacles. While moving, each robot senses the upcoming obstacles and generates a repulsive force which may lead him to get stuck in a local minimum of the overall potential functions. If this is the case, the robot detects that the repulsive force is contradicting the desired path to the leader, and finds a way to contour the obstacle by following the leader's track history. In this process, the robot breaks up all the formation links with the other team-mates that are in opposition to the obstacle repulsion force. In most simulations, the algorithm worked as expected, including situations where plain usage of the original algorithm would leave the followers trapped in local minima. Nevertheless, some problems were verified on the algorithm that was used to verify if an obstruction existed between a pair of vehicles. This verification, called the clearance condition, needs some additional work that may include a procedure to dynamically modify its configuration parameters, making them depend on other parameters like, *e.g.*, the distance between both vehicles, their velocity, angle displacement *w.r.t.* each other, *etc.*

In the presented simulation, several parameters associated with the formation framework were hand tuned. A systematic procedure, explanation, or criteria on how to adapt or automatically tune these parameters was not presented. Additional work on this area must naturally follow a more practical approach, with the definition and implementation of the respective user control interfaces. A next step to follow the set of simulations executed with Matlab can be, *e.g.*, simulating the framework on a more elaborate and realistic platform for robot simulations like Webots [1] or USARSim [2]. Figure 6.1 shows an example of the Webots simulation environment, obtained from their website. In these platforms a controller for each robot must be programmed. It is normally done in such a manner that, the program can be directly exported to a real robot, with just minor modifications.

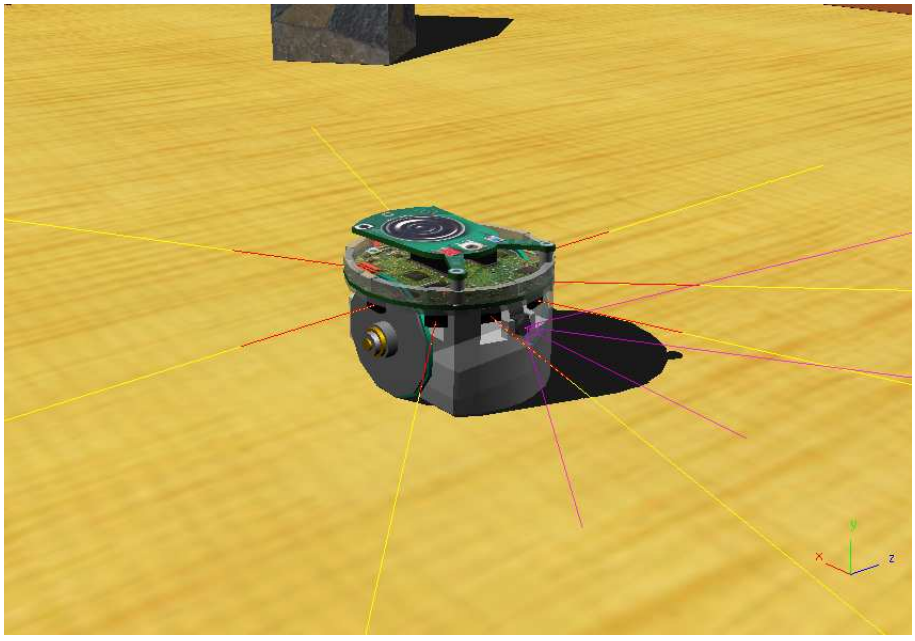


Figure 6.1: The Webots Simulation Environment.

The work in this thesis can gain a lot with multi-robot collaboration for localization, map building, cooperative sensing and every other multi-robot areas of research. Following the practical implementation of the ideas here presented, it will be interesting to study the impact that all the practical limitations will have on the formation structure namely, communication limitations, odometry and sensor errors, bounded

[1] http://www.cyberbotics.com/

[2] http://sourceforge.net/projects/usarsim

velocity and torque forces,*etc.* For future work, situations should be considered where each robot has only a partial and uncertain knowledge of the assumed information, to study the robustness of the algorithm to such a scenario.

To extend the line of work presented in this thesis, there are still several ideas to be explored. The following section presents other formation structures to be considered and, the last section, presents ideas on how to explore the application of NF in the formation, for obstacle avoidance and navigation.

6.1 Other Formation Structures

Additional theoretical studies on the formation framework can include, *e.g.*, describing the formation with a dynamic graph, study its stability, manage dynamic link breaking/creating in accordance with some predefined objectives, include orientation potentials to regulate the relative orientation that vehicles should assume *w.r.t.* each other, and the creation of other formation structures. Until now, every robot is connected to every other teammate within a certain area of influence. Alternately, links can be programmed so that vehicles are connected only to a certain subset of the formation members for attraction and repulsion. Only repulsive forces would be considered for the other members, to avoid inter-vehicle collisions. These collisions can also be avoided with orientation potentials, combined with a certain formation configuration to maintain vehicle apart. There is an extensive area to explore here that depends on the application requirements. Figure 6.2 shows, from the extensive set of possible structures, an example that represents a daisy chain link between formation groups. This is a type of hierarchical organization, with links preprogrammed for this configuration.

Using Rigid Links

Up until know, for proposed objectives, no need has yet been considered for any particular rigid formation structure. The controller for such structures, falls outside the scope of this thesis (see, *e.g.*, [42],[43],[44] and [45]). However, the idea of including such structures within the FF can open way for other interesting possibilities, *e.g.*, problems of cooperative manipulation, where a "rigid" formation may be necessary to transport a grasped object to a prescribed location. See, for example, [46]. Constraints can be imposed to the position and orientation that some vehicles should

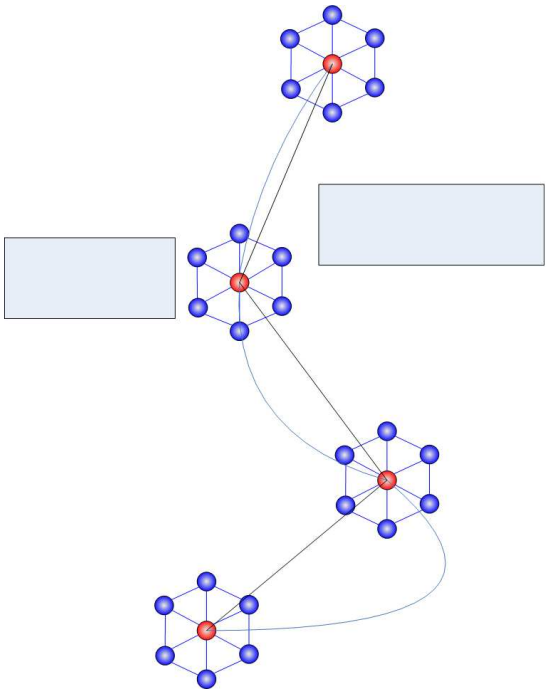


Figure 6.2: Daisy chain formation structure.

assume in the formation, creating a hybrid structure that can be useful for some applications. Figure 6.3 shows and example of such a structure where the green lines represent "rigid" links, *i.e.*, links that constrain the vehicles at every instant to a certain configuration *w.r.t.* each other, and blue lines, representing "soft" links associated with the IPs. The example showes a framework with three different groups. A

rigid structure can exist inside a group or between pairs of selected groups of vehicles. The latter will seem like adding a backbone to the formation. Other ideas can include, *e.g.*, linking two rigid formations with an IP, *etc*, as shown in figure 6.4. These are only some of the possibilities that can be explored. Future work will include the study of such structures, their dynamic graph representation, and stability analysis.

6.2 Using Navigation Functions

If somehow a NF could be constructed for the FF over $\mathfrak{F}$, it could be used on every robot to follow the leader without getting trapped (see Appendix A). The idea would be to exclude the leader IP from V_i and replace it by the corresponding NF. The

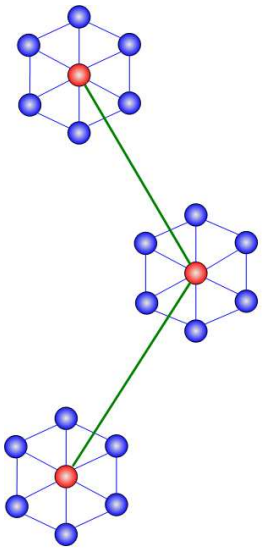


Figure 6.3: Formation that includes a "rigid" structure, represented by the green lines.

control force for each robot given by equation (3.4) will become:

$$\mathbf{u_i} = -\nabla_{\mathbf{q_i}} V_i - \mathbf{f_{vi}} - \mathbf{F}_{rep}(\mathbf{q_i}) - \nabla\varphi(\mathbf{q_i}) \tag{6.1}$$

where φ is constructed from the picture that each robot has of $\mathfrak{F}$ or, considering unlimited information sharing between every participating member of the FF, a global shared function, built upon the collective notion of $\mathfrak{F}$. Over the last years several references have been made in the literature to NFs, *e.g.*,[47] presents a method to compute NFs on complex shaped workspaces, using a Finite Element method for

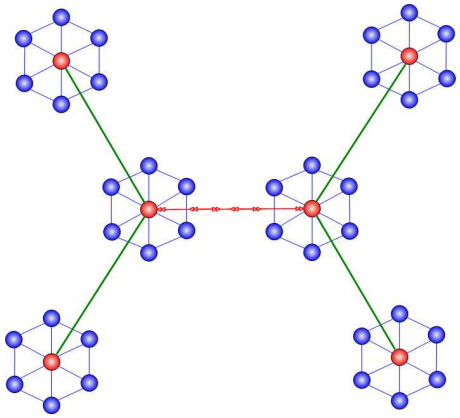


Figure 6.4: Two rigid structures (green lines), connected by an IP (red line).

potential field computation and in [48],[49] and [50], these functions are used for decentralized navigation, where a formation should reach a certain configuration or goal while avoiding collisions between its members.

The limitations that NFs present in practical applications is that their construction for arbitrary manifolds remains an art: each new model space W requires a handcrafted NF [51]. The basic steps for their construction namely, the description of W ; the computation of a diffeomorphism h between W and SW; the computation of a NF φ over SW; and the computation of a NF $\tilde{\varphi}$ over W , by transforming φ using h, are computationally expensive and difficult to implement. To complicate the process, the formation does not even have full knowledge of W . Even with this knowledge, just the process of describing its geometry with a Forest-of-Stars brings a computational overhead to the obstacle avoidance algorithm that does not coadunate with the proposed objectives.

Application Idea I

Every robot knows the position of all the other teammates and, since every robot avoids colliding with the obstacles, these positions are samples of $\mathfrak{F}$.

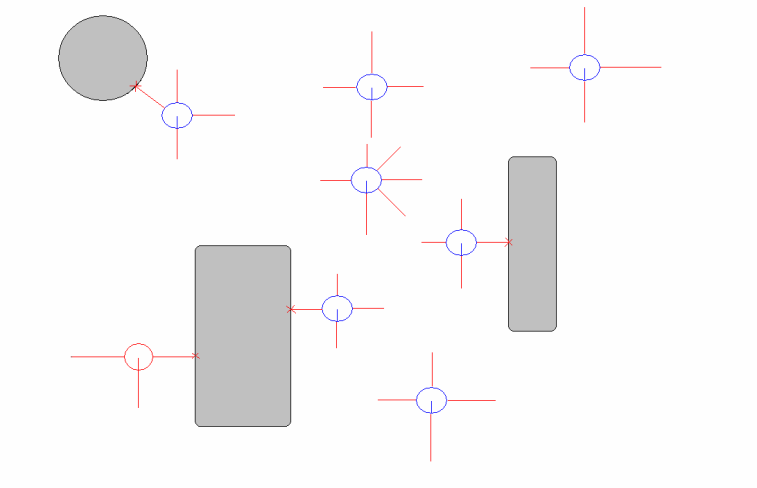


Figure 6.5: Formation in an obstacle field.

Figure 6.5 shows an example of a formation in an obstacle field. Every robot will be at a minimum distance from an obstacle and so, up to that limit, that space is part of $\mathfrak{F}$. The first application idea starts by each robot trying to describe $\partial\mathfrak{F}$ from information extrapolated from this set of points, information obtained from the

ODL, from the path followed by the leader and, possibly, information obtained using cooperative sensing and information exchange with other members of the formation. Figure 6.6 gives an idea of a possible extrapolated boundary $\partial\mathfrak{F}$ for the formation in figure 6.5.

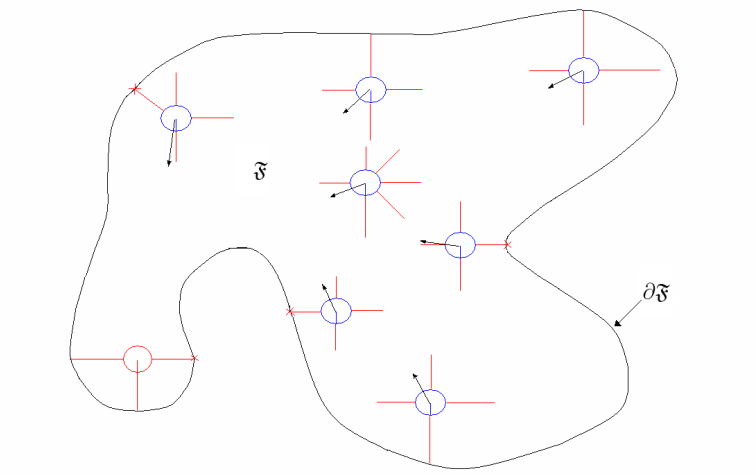


Figure 6.6: The extrapolated boundary $\partial\mathfrak{F}$.

The boundary will geometrically evolve at each iteration, with additional acquired information. If a mathematical description of $\mathfrak{F}$ could be arranged from the idea that a robot (or the formation) has of $\partial\mathfrak{F}$ and, if a diffeomorphism could be found to map it into a sphere, a NF could be built by proposition 1. Until now, no trivial solution was found for these tasks, thus the presented application remains a mere hypothesis.

Application Idea II

The second application idea consists of a tentative to directly describe W with a Sphere-World model. If the model suits for the description, a NF is available. ∂SW is defined by the minimum circumference that contains the entire formation and the path followed by the leader. While navigating, points are sampled using information from the ODL and, discs O_j with some predefined radius, are placed on points that correspond to obstacle boundaries. They must be placed in such a manner that they do not intercept each other. The leader's position is the destination point $\mathbf{q_d}$. Each robot will contain its own database of obstacles or, if no restrictions are applied for information exchange between robots, a shared database may be available for the entire formation. Figure 6.7 shows an example of this idea, where the green circles

represent the boundaries of the sphere obstacles.

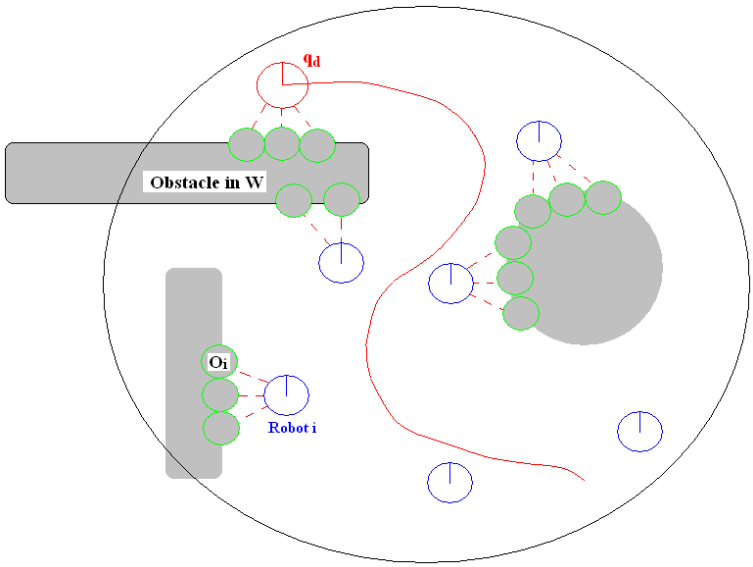


Figure 6.7: Direct application of the Sphere-World Model to the formation.

Bibliography

[1] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Auton. Robots*, 4(1):7–27, 1997.

[2] Pedro Fazenda and Pedro Lima. Non-holonomic robot formations with obstacle compliant geometry. *6th IFAC symposium on Intelligent Autonomous Vehicles*, September 3-5 2007.

[3] Edward A. Fiorelli. *Cooperative Vehicle Control, Feature Tracking and Ocean Sampling.* PhD thesis, Princeton University, Department of Mechanical & Aerospace Engineering, 2005.

[4] Brooks R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, pages 14–23, Mar 1986.

[5] G.Beni. The concept of cellular robotic system. *IEEE International Symposium on Intelligent Control*, pages 57–62, 1988.

[6] A. Mogilner, L.Edelstein-Keshet, L. Bent, and A.Spiros. Mutual interactions, potentials, and individual distance in a social aggregation. *J. Math. Biol.*, 47:353–389, 2003.

[7] Tucker Balch and Ronald C. Arkin. Motor schema-based formation control for multiagent robot teams. *Proceedings of the First International Conference on Multiagent Systems*, pages 10–16, 1995.

[8] Donald D. Dudenhoeffer and Michael P. Jones. A formation behavior for large-scale micro-robot force deployment. *Proceedings of the 2000 Winter Simulation Conference*, 1:972–982, Dec. 10-13 2000.

[9] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[10] Veysel Gazi and Kevin M. Passino. Stability analysis of social foraging swarms. *IEEE Transactions on Systems, MAN, and Cybernetics*, 34(1):539–557, February 2004.

[11] Michael M. Zavlanos and George J. Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics*, 23(4):812–816, August 2007.

[12] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2:2010– 2015, Dec 2003.

[13] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Stable flocking of mobile agents, part ii: Fixed topology. *Proceedings of 42nd IEEE Conference on Decision and Control*, 2:2016– 2021, Dec 2003.

[14] Mark Yamagishi. Social rules for reactive formation switching. Technical report, Univeristy of Washington, Electrical Engineering Department, Seattle.

[15] Andrew Howard, Maja J. M., and Gaurav S. S. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. *Proceedings of the 6th International Sysmposium on Distributed Autonomous Robotics Systems*, June 25-27 2002.

[16] Petter Ögren, Edward Fiorelli, and Naomi E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, Aug. 2003.

[17] Naomi E. Leonard and Edward Fiorelli. Virtual leaders,artificial potentials and coordinated control of groups. *Proceedings of the 40th IEEE Conference on Decision and Control*, 3:2968–2973, December 2001.

[18] Petter Ögren, Edward Fiorelli, and Naomi E. Leonard. Formations with a mission: Stable coordination of vehicle group maneuvers. *Proceedings of the 15 International Symposium on Mathematical Theory of Networks and Systems*, August 2002.

[19] Gabriel Hugh Elkaim and Michael Siegel. Extension of a lightweight formation control methodology to groups of autonomous vehicles. *The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, August 2005.

[20] Gabriel Hugh Elkaim and Michael Siegel. A lightweight control methodology for formation control of vehicle swarms. *Proceedings of the 16th IFAC World Congress*, July 4-8 2005.

[21] Gabriel Hugh Elkaim and Robert J. Kelbley. A lightweight formation control methodology for a swarm of non-holonomic vehicles. *IEEE Aerospace Conference*, March 4-11 2006.

[22] M.D Adams, H.Hu, and P.J.Roberts. Toward a real-time architecture for obstacle avoidance and path planning in mobile robots. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 584–589, May 1990.

[23] J.Borenstein and Y. Koren. Local obstacle avoidance for mobile robtos based on the method of artificial potentials. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 566–571, May 1990.

[24] R.B.Tilove. Real-time obstacle avoidance for fast mobile robots. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 572–577, May 1990.

[25] Y.Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1398–1404, April 7-12 1991.

[26] Frank E. Schneider and Dennis Wildermuth. A potential field based approach to multi robot formation navigation. *Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, October 2003.

[27] S.S. GE and Y.J. CUI. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3), November 2002.

[28] Herbert G. Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:3006– 3011, 26 April-1 May 2004 2004.

[29] Petter Ögren and Naomi E. Leonard. Obstacle avoidance in formation. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2:2492–2497, September 2003.

[30] Elon Rimon and Daniel E. Koditschek. Exact robot navigation using cost functions: The case of distinct spherical boundaries in e^n. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1791–1796, April 1998.

[31] Suranga Hettiarachchi and William M. Spears. Moving robot formations through obstacle fields. *Proceedings of the International Conference on Artificial Intelligence*, June 2005.

[32] G.Beni and J.Wang. Theoretical problems for the realization of distributed robotic systems. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1914–1920, 1991.

[33] Raúl Rojas and Alexander Gloye Förster. Holonomic control of a robot with an omnidirectional drive. *KI - Künstliche Intelligenz*, 20(2):12–17, 2006.

[34] Jean-Claude Latombe. *Robot Motion Planning - 1 edition.* Kluwer Academic, August 1991.

[35] Campion G., Bastin G., and D'Andrea-Novel B. Structural properties and classification of kinematic and dynamicmodels of wheeled mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1:462–469, May 1993.

[36] Bruno D. Damas, Pedro U. Lima, and Luis M. Custódio. *A Modified Potential Fields Method for Robot Navigation Applied to Dribbling in Robotic Soccer.* Springer Berlin / Heidelberg, 2003.

[37] Jonathan R. T. Lawton and Randal W. Beard. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, December 2003.

[38] J.B.Pomet, B.Thuilot, G.Bastin, and G.Campion. A hybrid strategy for the feedback stablization of nonholonomic mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 129–134, May 1992.

[39] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.

[40] Hassan K. Khalil. *Nonlinear Systems - 3rd Edition.* Prentice Hall, 2001.

[41] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E.Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* The MIT Press, 2005.

[42] Vito Trianni, Stefano Nolfi, and Marco Dorigo. Cooperative hole avoidance in a swarm-bot. Technical Report TR/IRIDIA/2004-22, Université Libre de Bruxelles, 2004.

[43] Xin Chen and Yangmin Li. Smooth formation navigation of multiple mobile robots for avoiding moving obstacles. *International Journal of Control, Automation, and Systems*, 4(4):466–479, August 2006.

[44] Jaydev P. Desai, James P. Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6), December 2001.

[45] João H.V.Costal and Pedro Lima. Formações de robots aéreos e terrestres. Technical report, Universidade Técnica de Lisboa/Instituto Superior Técnico.

[46] Peng Song and Vijay Kumar. A potential field based approach to multi-robot manipulation. *Proceeding of the IEEE International Conference on Robotics and Automation*, May 2002.

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

[48] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. Decentralized navigation functions for multiple robotic agents with limited sensing capabilities. *Journal of Intelligent and Robotic Systems*, 48(3):411–433, March 2007.

[49] Herbert G. Tanner and Amit Kumar. Formation stabilization of multiple agents using decentralized navigation functions. *Robotics: Science and Systems I*, June 2005.

[50] Herbert G. Tanner and Amit Kumar. Towards decentralization of multi-robot navigation functions. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4132– 4137, April 2005.

[51] N. Cowan. Composing navigation functions on cartesian products of manifolds with boundary. *Sixth International Workshop on the Algorithmic Foundations of Robotics*, 2004.

[52] Elon Rimon and Daniel E. Koditschek. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442, 1990.

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

[54] Daniel E. Koditschek. The application of total energy as a lyapunov function for mechanical control systems. *Contemporary Mathematics*, 97:131–157, 1989.

[55] Elon Rimon and Daniel E. Koditschek. The construction of analytic diffeomorphisms or exact robot navigation on star worlds. *Transactions of the American Mathematical Society*, 327(1):71–116, September 1991.

A
Navigation Functions

The main drawback of most potential field approaches is that, due to the possible presence of multiple local minima, convergence to a goal configuration $\mathbf{q_d}$ (in this thesis, the leader's position), is not guaranteed. If vehicles follow the negative gradient of the potential function from certain configurations, they will get trapped in areas that correspond to such local minima, either than $\mathbf{q_d}$. Rimon and Koditschek presented a method for designing potential functions that contain only one minimum, which is precisely at the goal. They show how to build such a function in a model space that they called *Sphere Worlds* (SW), and then presented procedures to generalize the idea to more elaborate spaces. Such a function is called a *Navigation Function* (NF) and all trajectories that start at any configuration $\mathbf{q_i} \in \mathfrak{F}$, that follow its negative gradient flow, will approach $\mathbf{q_d}$ without touching the boundary of the free space $\partial\mathfrak{F}$. A NF is formally defined by the following definition:

Definition:(from [52],[53]) A map $\varphi : E^n \rightarrow [0, 1]$, is a navigation function on a compact connected smooth manifold $\mathfrak{F} \subseteq W \subseteq E^n$ if it is:

1. Analytic on some open set containing $\mathfrak{F}$
2. Morse on $\mathfrak{F}$
3. Polar at $\mathbf{q_d}$, where $\mathbf{q_d} \in \mathfrak{F}$
4. Admissible on $\mathfrak{F}$

In mathematics, an analytic function is a function that is locally given by a convergent power series and is infinitely differentiable (also known as smooth, or C^∞). In a Morse function all critical points are non-degenerate. A critical point $\mathbf{q_c}$ is said to be non-degenerate if the Hessian of φ at $\mathbf{q_c}$ has a zero kernel. In other words,

all critical points are isolated and, when following the negative gradient, any random perturbation will destabilize saddles and maxima. The function is said to be polar on $\mathfrak{F}$ at $\mathbf{q_d}$ if it has exactly one minumum, at $\mathbf{q_d}$, and admissible if it is uniformly maximal on the boundary of the free space $\partial\mathfrak{F}$, *i.e.*,

$$\varphi(\mathbf{q_i}) = \begin{cases} = c & \text{for all} \quad \mathbf{q_i} \in \partial\mathfrak{F} \\ < c & \text{for all} \quad \mathbf{q_i} \in \mathfrak{F} \end{cases}$$

Exact Robot Navigation in Spherical Boundaries

The first step before extending the idea of NFs to general spaces is to build a NF on a model space SW, called a *Sphere World*. This section describes this workspace and how a NF is built for this particular space. The next section will describe how to build for other spaces, based on these results. The description starts by the following definition:

Definition:([30],[54]) In a world with spherical boundaries, the robot workspace $SW \in E^n$ is enclosed by a n-ball of radius ρ_0, centered at the origin of E^n and defined by

$$SW \triangleq \{\mathbf{q_x} \in E^n : q_x^2 \leqslant \rho_0^2\}$$

Every obstacle $O_j \in O \subseteq SW$ is an open ball of radius ρ_j centered at $\mathbf{q_j} \in SW$,

$$O_j \triangleq \{\mathbf{q_x} \in E^n : q_{xj}^2 \leqslant \rho_j^2\} \quad j = 1, \ldots, M$$

The free space is

$$\mathfrak{F} \triangleq SW - \bigcup_{j=1}^{M} O_j$$

Figure A.1 shows a example of a spherical bounded world in E^2, with 3 obstacles.

Navigation Function in SW

A navigation function in SW is a composition of three functions:

$$\varphi(\mathbf{q_x}) \triangleq \sigma_d \circ \sigma \circ \hat{\varphi}(\mathbf{q_x}) \tag{A.1}$$

Function $\hat{\varphi}$ is polar, almost everywhere Morse, and analytic: it attains a uniform height on $\partial\mathfrak{F}$, where $\hat{\varphi} \to \infty$. Function σ will map the co-domain of $\hat{\varphi}$, the interval $[0, \infty]$, into $[0, 1]$, where

$$\sigma(x) = \frac{x}{1+x} \tag{A.2}$$

resulting in a polar, admissible, and analytic function which is non-degenerate of $\mathfrak{F}$, except at the destination point $\mathbf{q_d}$. This is repaired by σ_d.

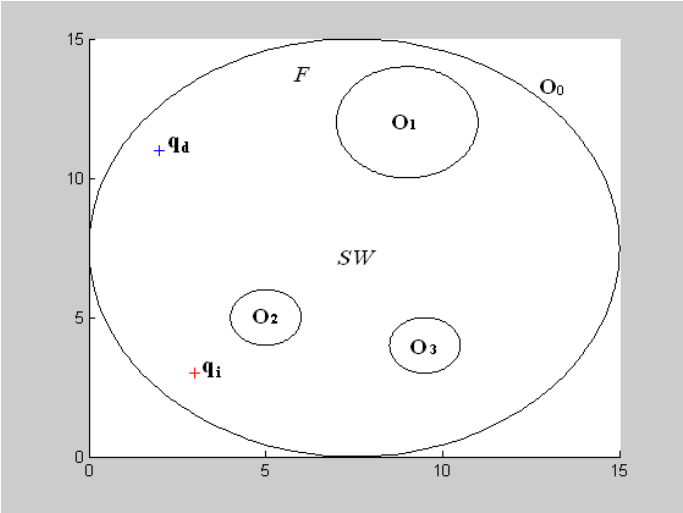


Figure A.1: Example of a Spherical Bounded World in E^2

Function $\hat{\varphi}$ is defined by a relation between two real valued maps γ and β, whose zero-levels, *i.e.*, $\gamma^{-1}(0)$ and $\beta^{-1}(0)$, are respectively the destination point $\mathbf{q_d}$ and $\partial\mathfrak{F}$. The function is given by:

$$\hat{\varphi}(\mathbf{q_x}) = \frac{\gamma(\mathbf{q_x})}{\beta(\mathbf{q_x})} \tag{A.3}$$

where $\gamma : \mathfrak{F} \to [0, \infty[$ is

$$\gamma(\mathbf{q_x}) = \gamma_d^k(\mathbf{q_x}) \quad k \in \mathbb{N}; \quad \gamma_d(\mathbf{q_x}) = q_{xd}^2,$$

a function directly proportional to the euclidian distance between $\mathbf{q_x}$ and $\mathbf{q_d}$.

$\partial\mathfrak{F}$ are areas to be avoided and are associated with the boundary of SW and every obstacle $O_j \in O$. $\beta : \mathfrak{F} \to [0, \infty[$ is a cost function associated with these areas and is given by

$$\beta(\mathbf{q_x}) = \prod_{j=0}^{M} \beta_j(\mathbf{q_x})$$

where $\beta_0(\mathbf{q_x})$ and $\beta_j(\mathbf{q_x})$ are cost functions associated with the distance to the respective boundaries ∂SW and ∂O_j, given by

$$\beta_0(\mathbf{q_x}) = \rho_0^2 - q_x^2\,; \quad \beta_j(\mathbf{q_x}) = q_{xj}^2 - \rho_j^2 \quad j = 1, \ldots, M$$

Due to the parameter k in $\hat{\varphi}$, the destination point $\mathbf{q_d}$ is a *degenerate critical point*. To counteract this effect, the "distortion" $\sigma_d : [0, 1] \to [0, 1]$,

$$\sigma_d(x) = x^{\frac{1}{k}} \quad k \in \mathbb{N}, \tag{A.4}$$

is introduced, to change $\mathbf{q_d}$ to a non-degenerate critical point.

Valid Workspace

For SW to be a *valid workspace* the following points must be verified:

1. $\mathbf{q_d}$ is in the interior of $\mathfrak{F}$,

$$\beta_i(\mathbf{q_d}) > 0 \quad i \in \{0, \ldots, M\}$$

2. All the obstacles are contained in the interior of SW,

$$\beta_0(\mathbf{q_i}) > 0 \quad \text{and} \quad q_i + \rho_i < \rho_0 \quad i \in \{0, \ldots, M\}$$

3. The obstacles do not intersect,

$$q_{ij} > \rho_i + \rho_j \quad i, j \in \{0, \ldots, M\}$$

If SW is a valid workspace, then there exists a positive integer $N \in \mathbb{N}$, such that for every $k \geq N$

$$\varphi \triangleq \sigma_d \circ \sigma \circ \hat{\varphi} = \left(\frac{\gamma_d(\mathbf{q_i})^k}{\gamma_d(\mathbf{q_i})^k + \beta} \right)^{\frac{1}{k}} \tag{A.5}$$

is a navigation function on $\mathfrak{F}$ (see proof in [52]). This function has zero value at $\mathbf{q_d}$ and continuously increases as a configuration moves away from the goal. As k increases, the influence of the goal attraction will tend to prevail over the repulsive influence of the obstacles. Critical points either than $\mathbf{q_d}$ "gravitate" toward the obstacles and will eventually become unstable saddle points.

Figure A.2 shows the respective 2D and 3D contour plots of equation (A.5), for the world shown in figure A.1, using $k = 2.0$, $k = 2.5$ and $k = 2.9$.

Star Worlds and Extension to Geometrically Complicated Spaces

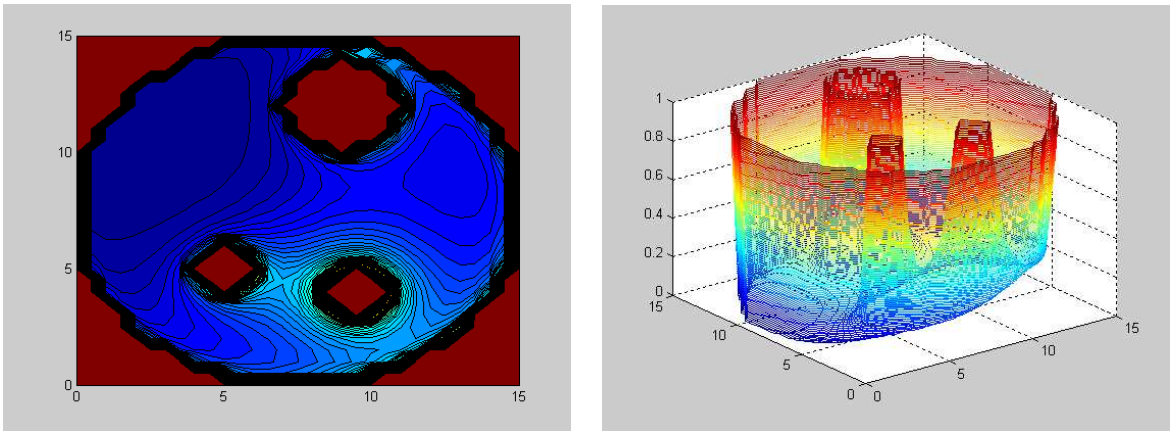
The Sphere-Space SWcan serve as a model space to any space W that is diffeomorphic to it. Finding a navigation function in this space means finding the diffeomorphism relating both spaces, considering the following proposition:

Proposition 1:([55]) Let $\varphi : SW \to [0, 1]$ be a *navigation function* on SW, an let $h : W \to SW$ be analytc. If h is an analytic diffeomorphism, then

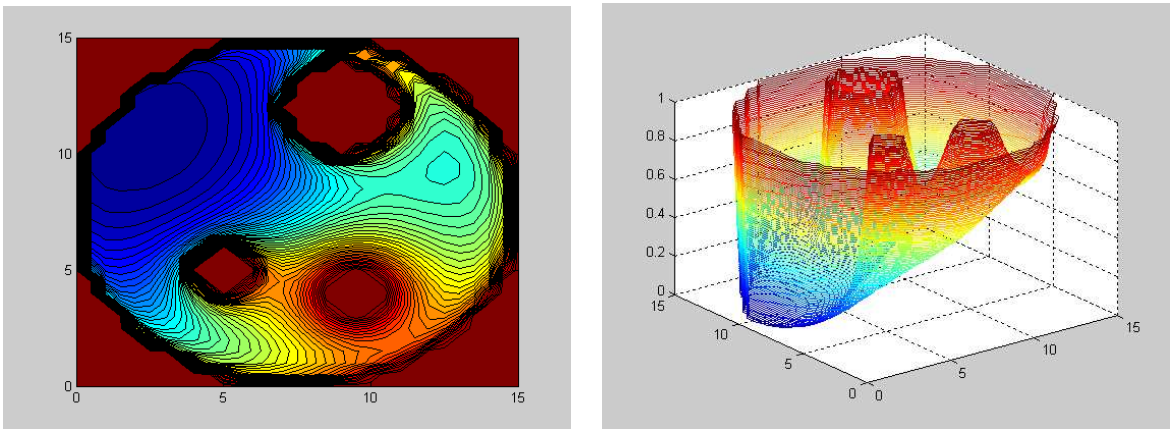
$$\tilde{\varphi} = \varphi \circ h$$

is a navigation function on W .

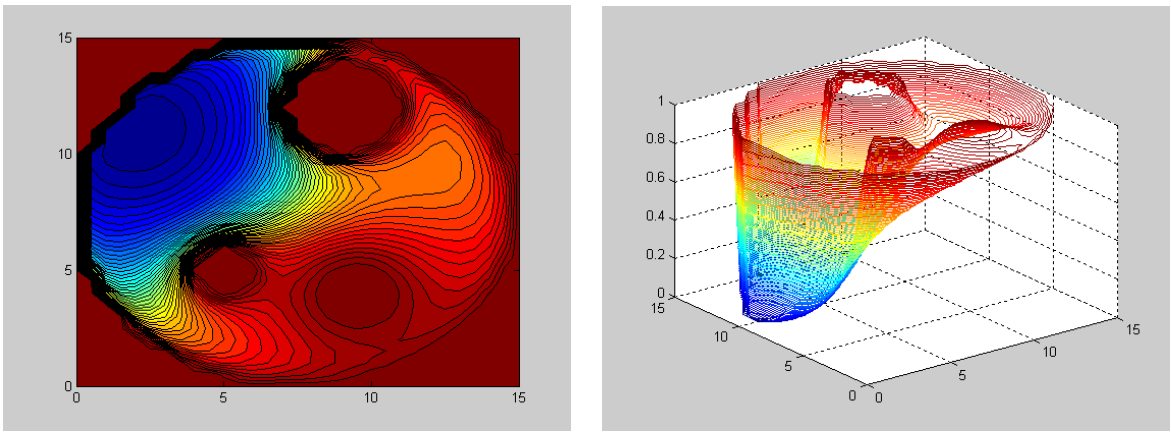
A NF can be found for any space if a suitable diffeomorphism, h, can be found relating both spaces. Rimon and Koditschek present an example by using a SW topological equivalent space call the *Star Worlds* (STW). Star-Shaped sets extend convex sets and are geometrically more "expressive" than spheres. The authors show how to map the boundary of a star onto the boundary of a respective disk, the star's



(a) $k = 2.0$



(b) $k = 2.5$



(c) $k = 2.9$

Figure A.2: 2D and 3D contour plot of φ, for the world shown in figure A.1, using different k values.

interior into the disk's interior and the star's outside to the disk's outside, by means of Star-to-Sphere transformations called *Translated scaling maps*. The diffeomorphism h, mapping the STW into SW, will be defined by a linear combination of such transformations. Figure A.3 shows an example. The outer sphere is chosen sufficiently large so that it contains the outer boundary of the star worlds.

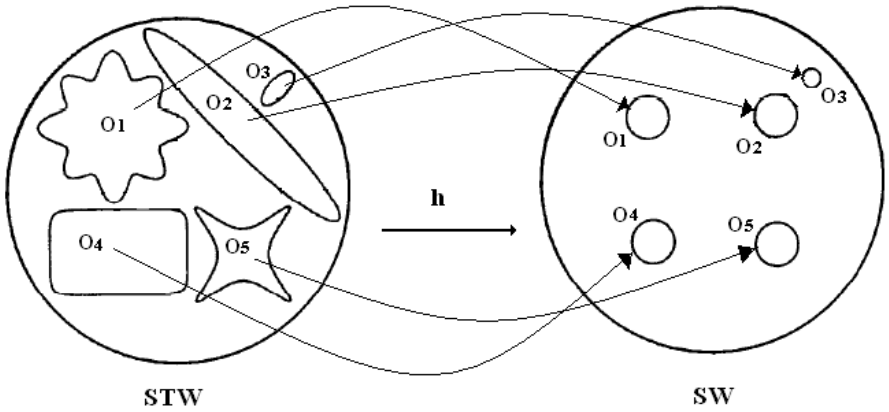


Figure A.3: The diffeomorphism h maps the Star-Space STW to the Sphere-Space SW.

To increase the geometric expressiveness of the obstacle representation and to generalize to more realistic scenarios, obstacles can be represented by a finite union of overlapping stars called a *Tree-of-Stars*. A world W with several disjoint obstacles is called a *Forest-of-Stars*, that can also can be mapped to the model sphere-space. Figure A.4 shows and example of a Forest-of-Stars, composed by 3 trees.

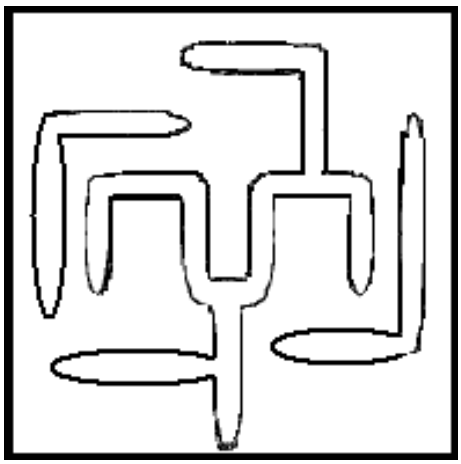


Figure A.4: Forest-of-Stars, composed by 3 Trees-of-Stars.