

NONSMOOTH MULTI-AGENT NAVIGATION FUNCTIONS

by

Adithya Boddu

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Winter 2012

© 2012 Adithya Boddu
All Rights Reserved

**NONSMOOTH MULTI-AGENT NAVIGATION
FUNCTIONS**

by

Adithya Boddu

Approved: _____
Herbert G. Tanner, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Anette M. Karlsson, Ph.D.
Chair of the Department of Mechanical Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Interim Dean of the College of Engineering

Approved: _____
Charles G. Riordan, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGEMENTS

I express my deepest gratitude to my adviser, Dr. Herbert Glenn Tanner, who has been extremely supportive of me all through my research. I thank him for all his unflinching cooperation, guidance and patience while I was working with him. I attribute my research skills especially in math, to him.

I thank the committee members Dr. Loannis Poulakakis and Dr. Ryan Zurawski for their comments and suggestions, which helped me to improve this work.

I would also like to thank my lab-mates in the Cooperative Robotics Laboratory for their suggestions and the productive discussions that I had with them. Special thanks to Nicholas Lacock for his inputs in the C code that I have written for the formation stabilization of the *Corobots*. I thank Shridhar, Varsha and Luis for their timely help.

I am greatly indebted to my family for their unconditional support, and the faith they have shown in me. My mom B. Shobha Rani has been a constant source of motivation throughout. I dedicate this thesis to my late dad Mr. B. M. Prabhu who inspired me into pursuing higher education with master's.

I must also thank my friends Vimal, Mayur, Vignesh, Praveen, Hemanth, and Vishnu who have made themselves available for their help and cooperation in numerous ways.

Lastly, I express my regards to all of those who supported me in any respect during the completion of my work. This work was supported by NSF under grant 0913015.

TABLE OF CONTENTS

LIST OF FIGURES	viii
LIST OF TABLES	xii
ABSTRACT	xiii
1 INTRODUCTION	1
1.1 The Motion Planning Problem - Description	1
1.1.1 What is Motion Planning	1
1.1.2 Path Planning	1
1.1.3 Trajectory Generation	2
1.1.4 Robot Control	3
1.2 Motivation For This Work	3
1.3 Contribution of this work	4
1.4 Thesis Overview	4
2 TECHNICAL BACKGROUND AND LITERATURE SURVEY	6
2.1 Introduction	6
2.2 General Navigation Problem Description [29]	7
2.3 General Path Planning Approaches	7
2.3.1 The Roadmap Method	7
2.3.1.1 Visibility Graph	8
2.3.1.2 Voronoi Diagram	8
2.3.2 The Cell Decomposition Method	9
2.3.3 The Potential Field Method	9

2.3.4	Combination of Cell Decomposition and Potential Field Method	11
2.4	Motion planning for groups of robots	12
2.4.1	Centralized Navigation Scheme	12
2.4.2	Decentralized Navigation Scheme	13
2.4.3	Approaches based on formal languages	14
2.5	Challenges	15
3	A CLOSER LOOK AT EXISTING METHODS	17
3.1	Single agent navigation function construction	18
3.2	The goal function	20
3.3	The obstacle function	21
3.3.1	Analytical switches	22
3.4	Decomposition of the workspace \mathcal{F}	23
3.5	Why φ is a navigation function	24
3.6	Attempts at extending to multiple agents	25
3.7	Challenges	28
4	NONSMOOTH ANALYSIS AND THE DISTANCE FUNCTION	29
4.1	Generalized directional derivative	29
4.2	Generalized gradient	30
4.3	Generalized gradient-pointwise minima	30
4.4	Filippov Solutions	32
4.5	The distance function and its properties	35
4.6	The distance function in the new multi-agent navigation function	35
4.7	Example: generalized gradient of pointwise minima	38
5	A NEW CONSTRUCTION FOR MULTI-AGENT NAVIGATION	42
5.1	Construction	42
5.2	The formation graph	43
5.3	The goal function	44
5.4	The obstacle function	45

5.5	Properties of the goal and obstacle functions	46
5.5.1	Goal function properties	46
5.5.2	Obstacle function properties	46
6	PROOF OF CORRECTNESS	49
6.1	When the distance function is differentiable	49
6.1.1	The destination is non-degenerate	49
6.1.2	There are no critical points on the workspace boundary $\partial\mathcal{F}$	52
6.1.3	The critical points can be pushed toward collision configurations	52
6.1.4	Critical points close to collision configurations are not minima	53
6.1.5	The navigation function is Morse	57
6.2	When the distance function is not differentiable	61
6.2.1	Extension of Proposition 4	62
6.2.2	Extension of Proposition 6	62
6.2.3	Extension of Proposition 7	63
6.2.4	Extension of Proposition 8	63
6.2.5	Extension of Proposition 9: The navigation function is Morse	65
6.3	Proof of convergence of φ	68
6.4	Proof of convergence using Ryan's invariance principle	70
7	NUMERICAL AND EXPERIMENTAL VALIDATION	72
7.1	Numerical Results	72
7.1.1	Assessment of the critical points	73
7.1.1.1	Nature of the critical points - A comparison between the traditional construction and the new construction	73
7.1.2	Simulation of agent paths	73
7.2	Experimental Validation	79
7.2.1	Corobots	79

7.2.2	Vicon Motion Capture System	81
7.2.3	The formation controller	82
7.2.4	Experimental results	84
8	CONCLUSIONS AND FUTURE WORK	89
	BIBLIOGRAPHY	91
 Appendix		
	DEFINITIONS	96

LIST OF FIGURES

1.1	General Motion Planning	1
2.1	Visibility Graph (<i>cf.</i> [29])	8
2.2	Cell Decomposition (<i>cf.</i> [29])	10
3.1	A figurative description of the decomposition of the workspace \mathcal{F} into different regions. \mathcal{F}_0 represents the neighborhood of the obstacles. \mathcal{F}_1 represents the neighborhood of the boundary of workspace $\partial\mathcal{F}$. q_d denotes the destination configuration in the free workspace \mathcal{F}_2	24
4.1	Three 3-D slices of an inter-agent distance function defined in a four-dimensional space. The collision configurations between agents 1 and 2 are marked by the cylinder that contains the (hyper)line $x_{12} = y_{12} = 0$. In the three-dimensional slices where the dimension y_{13} is not pictured, the collision configurations between agents 1 and 3 are shown as the “thick” hyperplane passing through the origin on the $y_1 - y_3 = 0$ slice. Note the diagonal cylinder with axis on the x_{13} - x_{12} plane: this represents collisions between agents 2 and 3 (although x_{23} and y_{23} are not mapped). This diagonal collision region expresses the fact that when $q_{12} = q_{13}$, agents 2 and 3 overlap; at the slice where $y_{13} = 0$, therefore, and on the plane where $y_{12} = 0 = y_{13}$, the diagonal line $x_{12} = x_{13}$ maps configurations where all three agents have the same y coordinate, and agent 2 is on top of agent 3. These three graphs illustrate that pairwise obstacle functions (i.e., collision between 1 and 2, or collision between 1 and 3) define regions in the relative position space which are not isolated, and irrespectively of the agents’ volume the origin of this space will always be a point common to all regions.	37

4.2	<p>The $\min\{ x , y \}$ function and its generalized derivative. Figure 4.2(a) shows that the function has minima along the x and y axes. Along the lines $x = y$ and $x = -y$ the function is not differentiable. Figure 4.2(b) illustrates why the generalized gradient along the $x = \pm y$ lines does not contain zero: $(0, 0) \notin \text{co}\{(0, 1), (1, 0)\}$! There is no positive λ for which $(0, 0) = \lambda(0, 1) + (1 - \lambda)(1, 0)$.</p>	39
5.1	<p>Variation in the shape of the goal function γ with the tuning parameter κ. The destination for the model goal function $((x - 1)^2 + (y - 1)^2)^\kappa$ is $(1, 1)$ Fig. 5.1(a) shows the graph for $\kappa = 0.3$, where the shape of the function is more sharpened towards the destination. Fig. 5.1(b) shows the same for $\kappa = 0.6$. For higher κ, the shape of γ looks smooth towards the destination as can be seen in Fig. 5.1(c)</p>	47
7.1	<p>Numerical computation of critical points in a traditional multi-agent navigation function as given by (3.9). Agent 1 is assumed at the center, and the relative position of the other two agents when φ has a saddle is marked by a pair of small circles. Points $(0.2, 0.2)$ and $(0.3, 0.3)$ mark the desired configuration. For small values of κ a continuum of critical points appears; it later disappears as the parameter increases beyond a certain threshold.</p>	74
7.2	<p>Numerical computation of critical points in the proposed multi-agent navigation function given by (5.1)-(5.2)-(5.3)-(5.4). Agent 1 is assumed at the center, and the relative position of the other two agents when φ has a saddle is marked by a pair of small circles. Points $(0.2, 0.2)$ and $(0.3, 0.3)$ mark the desired configuration. For small values of κ a ring of critical points appears and then disappears as the parameter increases beyond a certain threshold.</p>	75

7.3	<p>Simulation of a formation control maneuver where all agents have to switch positions, starting within the region of attraction of a saddle point which eventually traps them. In Figure 7.3(a) the continuous curve shows the evolution of $x_1 - x_2$, the dashed curve shows that of $y_1 - y_2$, the dot-dashed curve depicts $x_2 - x_3$, and the thick dotted one gives $y_2 - y_3$. Figure 7.3(b) reveals that the agents are not capable of minimizing their goal function γ. In Figure 7.3(c) the continuous path corresponds to that of agent 1, the dashed curve to agent 2, and the dot-dashed to agent 3. The initial positions of the agents are marked with a dot and their corresponding numeral, while the small arrows at the end of the paths denote the final positions for the agents at the end of the time period of simulation. Figure 7.3(c) shows that the agents stay on the same line they started at as they converge to the saddle. The chattering that occurs at steady-state is due to the fixed integration step of the numerical solver.</p>	77
7.4	<p>Simulation of a formation control maneuver where all agents have to switch positions, starting <i>close</i> but not in the attraction region of a saddle. The notation is the same as that in Figure 7.3. In the initial configuration, $q_{12}(0) = (-0.305, -0.3)$ m and $q_{23}(0) = (0.1, 0.1)$ m. In Figure 7.4(a) the continuous curve shows the evolution of $x_1 - x_2$, the dashed curve shows that of $y_1 - y_2$, the dot-dashed curve depicts $x_2 - x_3$, and the thick dotted one gives $y_2 - y_3$. Figure 7.4(b) indicates that the desired formation is reached since the goal function γ converges to zero. Figure 7.4(c) shows the paths of the three agents as they switch positions to reconfigure themselves into the desired formation.</p>	78
7.5	<p>Simulation of a formation control maneuver where all agents have to switch positions, starting in the neighborhood of a saddle configuration. The notation is the same as that in Fig. 7.3. The initial configuration, $q_{12}(0) = (-0.405, -0.4)$ m and $q_{23}(0) = (0.2, 0.2)$ m. In Fig. 7.5(a) shows with continuous line the evolution of $x_1 - x_2$, with dashed curve that of $y_1 - y_2$, with dot-dashed curve that of $x_2 - x_3$, and with thick dotted one the trajectory of $y_2 - y_3$. Figure 7.5(b) shows the paths of the three agents as they switch positions to reconfigure themselves into the desired formation, where it can be seen that their final position is different from that of the case depicted in Fig. 7.4.</p>	80

7.6	Initial and final configurations for the 6-agent simulation test . . .	81
7.7	Simulation of a 6-agent formation control maneuver where the agents switch positions in accordance with Fig. 7.6. The agents begin on a 45° straight line in the neighborhood of a saddle configuration. The relative distances between successive agents is $(0.1, 0.1)$. Figure 7.7(a) shows the agent paths, with unprimed numbers representing the initial configuration and primed numbers the final configuration. Figure 7.7(b) shows the evolution of the relative x-distances between the agents. Figure 7.7(c) shows the evolution of the relative-ydistances between the agents.	82
7.8	Experimental results for the line formation of three agents from almost any initial configuration	86
7.9	Experimental results showing the switching of relative positions of three agents in a straight line.	87
7.10	Experimental results showing three agents switching in a triangle.	88

LIST OF TABLES

7.1	Computation Time	79
-----	----------------------------	----

ABSTRACT

This work presents a path planning methodology for designing centralized control laws which can provably steer a group of robotic agents to fall into a free-floating formation of arbitrary shape, while following collision free trajectories.

The planning scheme is based on the concept of navigation functions, a special type of artificial potential functions without local minima. In this thesis, we describe how this idea can be generalized from its original formulation for single-robot systems, to multi-robot formations. Further, we indicate why existing solutions that have appeared in literature, although potentially functional, have not unequivocally established the non-degeneracy of undesirable critical points of the potential function. The problem is therefore reconsidered here, and a new (nonsmooth) potential function is proposed along with the associated control strategy.

We show that the new construction of the multi agent navigation function allows analytic proofs for the convergence of the closed loop system and the non-degeneracy of the critical points of the underlying potential field.

Chapter 1

INTRODUCTION

1.1 The Motion Planning Problem - Description

The Motion Planning Problem is traditionally divided into three tasks: *Path Planning*, *Trajectory Generation* and *Robot Control*.

1.1.1 What is Motion Planning

The process of steering a mobile agent or a group of agents to a specified destination/formation without any collisions and in a finite amount of time is the basic idea of motion planning.

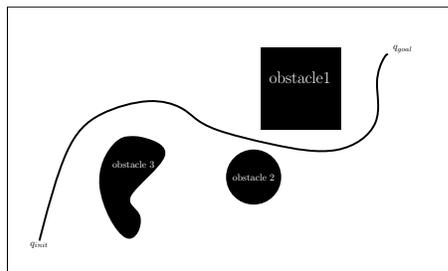


Figure 1.1: General Motion Planning

1.1.2 Path Planning

Definition 1 (Path). *A path is a continuous curve of robot positions in the workspace, not necessarily being a function of time.*

Definition 2 (Trajectory). *A continuous function τ from $[0, 1]$ into the workspace such that $\tau(0) = q_i$ and $\tau(1) = q_d$.*

In *path planning*, the objective is to design an algorithm that generates a path in the workspace based on the geometric data available without considering the dynamics of the robot. *Path planning*, hence, constitutes the problem of planning collision-free paths in the workspace. The initial and final configurations of the robot are assumed to be fixed and the goal is to find a collision-free path that connects them. The robot is assumed to be a free-moving object in space capable of moving in any direction without any constraints except the obstacles in the environment. The basic path planning problem can be viewed as follows

Definition 3 (Path Planning). *Assuming the robot to be omnidirectional,¹ find an algorithm that moves the robot from the start point to the destination avoiding collision with the obstacles.*

There are various methods of finding a solution to the basic path planning problem which are described in Section 2.3.

1.1.3 Trajectory Generation

The path generated by the path planning algorithm, may be infeasible for the robot due to the constraints (minimum turning radius, maximum velocity possible, etc.) on the robot. This path needs to be converted into a trajectory which can be followable by the robot.

In the basic path planning problem the robot was assumed to be capable of moving in any direction from any given location in the workspace. Apart from the geometric constraints imposed by the obstacles on the agents, the constraints are classified as holonomic and nonholonomic constraints. Holonomic constraints are

¹ A robot is said to be omnidirectional if it can move in any direction.

the velocity/acceleration constraints which can be integrated to yield constraints on the position variables and nonholonomic constraints are those which cannot be integrated to yield constraints on the position variables). In the presence of such constraints on the robot, a sequence of positions for the robot as a function of time is to be generated.

1.1.4 Robot Control

In *Robot Control* we develop the control inputs that drive the robot along the trajectory generated in the *trajectory generation* task.

The generated trajectory needs to be followed by the robot closely with a feedback control. A suitable feedback controller is designed for this purpose so that the required inputs to the robot are obtained from this controller.

1.2 Motivation For This Work

The main motivation for this work is to cater to the need to navigate multiple agents in a workspace. Formation control of multiple agents allows the distribution of complex tasks among the agents thus increasing the efficiency of the system. Also, since multiple agents could perform a task in lesser time than would a single agent, as in the case of exploration of unknown environments with multiple agents in a specific formation, the need to develop control strategies for the navigation of multiple agents has evolved. As a result, the need to control multiple robots at the same time has gained momentum. While open-loop planning approaches lack the required robustness, closed-loop approaches guarantee the convergence even with some degree of uncertainty. Closed-loop approaches such as potential field based navigation functions have thus gained importance in motion planning. A number of attempts have been made to extend the potential field based single-agent navigation to multiple agents based on the seminal work by Rimon and Koditschek [28].

Although, these approaches seem to perform well in tests (numerical and experimental) the proof of correctness has not been fully established in a mathematically rigorous way. The main motivation for this work is to complete the correctness proof of the potential field based multi-agent navigation function; it turns out that an alternative design is needed.

1.3 Contribution of this work

The main contribution of this work lies in identifying the limitations in existing approaches to extending the potential-field based single agent navigation function to a multi-agent setting, and overcoming these limitations by proposing a new framework for the multi-agent navigation function.

Contrary to the claims therein, in existing literature the proof of convergence for the multiple robots in formation control is based on assumptions that are violated. Although the the mathematical mishap does not manifest itself easily in numerical testing, it does not allow one to establish all the desired properties of the potential field. We propose a new construction for the multi-agent navigation function, and a proof of correctness for all the required properties.

1.4 Thesis Overview

The rest of the thesis is organized as follows.

Chapter 2 provides technical background that forms the basis for the subsequent chapters. The general motion planning problem is described and the current state of research in multi-agent navigation control is discussed. Existing planning approaches are presented along with their limitations.

In Chapter 3 we narrow down our focus to the navigation function introduced by Rimon and Koditschek in [28]. We discuss its construction and the challenges encountered in extending the same to multiple agents.

In Chapter 4 we present few results from nonsmooth analysis, which are later utilized in demonstrating the proof of correctness of the new construction of the navigation function. We discuss the properties of the (nonsmooth) distance function in detail and describe its differentiability properties. The generalized gradient of the distance function is discussed here and its role in the new navigation function.

In Chapter 5 we present the new construction for the navigation function and its properties. In Chapter 6 we demonstrate the proof of correctness of the new multi-agent navigation function. In Chapter 7 we present the numerical and experimental results associated with the new construction. The experimental setup for the simulation is described. Computer simulation results for the 3-agent and 6-agent cases are shown, while the experimental results for the 3-agent case with the *Corobots* are presented.

Chapter 2

TECHNICAL BACKGROUND AND LITERATURE SURVEY

This chapter is intended to give a review of the state-of-the-art in path-planning, emphasizing the methods on navigation function based formation control stabilization. We shall have a thorough look at some of the existing methods for path planning, their advantages and disadvantages.

2.1 Introduction

With increase in the efficiency and performance of portable computational devices, the focus in motion planning has shifted to the theoretical aspects of planning and control. The study of theory bridges the gap between the fast hardware architecture and the design criterion behind motion planning. The algorithms that go into planning can afford to be complex with the advent of such hardware. Several design techniques have appeared in literature, and a partial list can include [40], [6], [36], [47], and [23]. The primary focus of this work is on a specific methodology that generalizes a potential field approach for single robots, to groups of robots. This methodology promises algorithmic completeness, a combined motion planning and path following control construction that employs feedback, and a closed loop system that enjoys global convergence guarantees. It is based on an artificial potential function which can be tuned so that it does not have any local minima, known as a *navigation function* [38]. We shall have a look at the existing work on multi-agent navigation and formation control.

2.2 General Navigation Problem Description [29]

Let \mathcal{A} be an agent moving in a subset \mathcal{W} of the Euclidean space \mathbb{R}^N , where N is the dimension of the workspace \mathcal{W} and let $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m$ represent the obstacles in the workspace \mathcal{W} . Assuming that the location and shape of $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m$ is accurately known and that \mathcal{A} is omnidirectional, meaning that there are no kinematic constraints on \mathcal{A} , we can define the problem as

Given an initial position and orientation and a goal position and orientation of \mathcal{A} in \mathcal{W} , find an algorithm that defines the course of travel of \mathcal{A} from the initial location to the destination avoiding contact with \mathcal{B}_i 's and report failure if no such path exists.

2.3 General Path Planning Approaches

There are a number of methods for solving the path planning problem, however not all solve the problem in full generality, *i.e.* some *planners* are not complete,¹ and some have the problem of getting stuck in non-optimal configurations. We look at some of the traditional planning methods and their limitations.

2.3.1 The Roadmap Method

A roadmap is the set of curves in the free space that are constructed based on a specific topological feature. It is a data base of all the possible collision-free paths between the starting point to the destination. The roadmap \mathbf{R} can be written as,

$$\mathbf{R} = \bigcup \tau_i$$

Different methods of constructing roadmaps are *visibility graph* method, *retraction* method, and *silhouette* method. After the roadmap is constructed using any of these methods, the start and end points are connected to the roadmap. The path from the start point to the destination is therefore a concatenation of the path

¹ A planner is said to be complete if it finds a solution when one exists and reports a failure otherwise.

connecting the goal to the roadmap, the path contained within the roadmap and, the path that connects the roadmap to the destination configuration. A number of methods to construct roadmaps have been proposed. We describe two of them, the visibility graph and Voronoi diagram methods.

2.3.1.1 Visibility Graph

This method is limited to the configuration spaces which are subsets of \mathbb{R}^2 where the obstacles are all polygonal. The visibility graph is an undirected graph whose nodes are the initial and goal configurations q_{init} and q_{goal} , and the obstacle vertices. The visibility graph \mathbf{VG} edges consist of straight lines joining all the nodes which do not intersect with the obstacles. The shortest path between the q_{init} and q_{goal} configurations is the solution to the path planning problem.

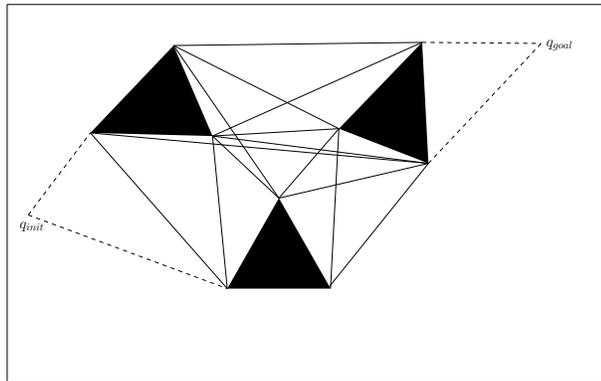


Figure 2.1: Visibility Graph (*cf.* [29])

2.3.1.2 Voronoi Diagram

In a Voronoi diagram the curves are so constructed that the distance of any point in the diagram is always equidistant from the two closest obstacles. The

advantage of the Voronoi diagram is that, the robot is always at the farthest distance from the nearest obstacles. A Voronoi diagram can be defined as

$$\mathcal{V} = \bigcup \mathcal{F}_i$$

where,

$$\mathcal{F}_i = \{q \in \mathcal{Q}_{free} | d_i(q) \leq d_h(q) \forall h \neq i\}$$

\mathcal{Q}_{free} is the obstacle-free space defined as $\mathcal{Q}_{free} = \mathcal{W} - \bigcup \mathcal{QO}_i$

\mathcal{QO}_i is the i^{th} obstacle.

$d_i(q)$ is the distance of the point q to the obstacle \mathcal{QO}_i

Unlike the visibility graph approach, this method is not restricted to polygonal obstacles.

2.3.2 The Cell Decomposition Method

The cell decomposition method is the most commonly used method for motion planning. The workspace is divided into simple regions called cells, so that a path between any two adjacent cells is generated. Two cells are said to be adjacent if they share a common boundary. An adjacency graph using these cells is thus constructed, in which each node represents a cell and a link connecting the nodes corresponds to the common edge shared between those cells.

Once the adjacency graph is constructed, the cells in the graph that contain the start and end points are located, and the graph is searched for a feasible path between those nodes.

2.3.3 The Potential Field Method

In the potential field method, the workspace is modeled as a landscape of mountains and valleys where the obstacles are represented by mountains, free space as valleys, and the destination is the point with the lowest elevation.

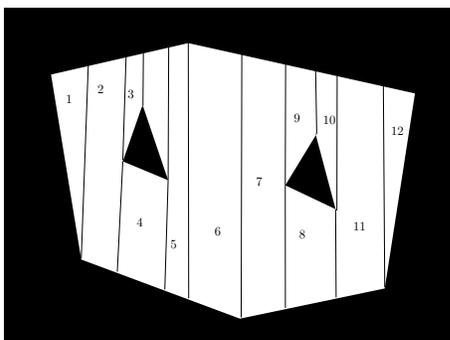


Figure 2.2: Cell Decomposition (*cf.* [29])

In the potential field approach, the primary idea is to generate artificial attractive and repulsive forces within the workspace and create controllers that respond to these forces guiding the robot to the target destination while avoiding obstacles. The obvious choice was to attribute attractive field to the destination and repulsive field to the obstacles. Due to continuous interaction between the attractive and repulsive forces, the potential field method thus provides a solution to the path planning problem that allows the use of feedback for online planning.

In classical potential field methods the resultant artificial force acting on the robot at any point is the sum of attractive and repulsive fields. Ignoring the dynamics, we can view the potential field as a velocity reference at each point.

The total potential field is expressed as

$$\mathcal{U}_{total} = \mathcal{U}_{att} + \mathcal{U}_{rep} .$$

The first instance of the usage of potential field technique in robotics is attributed to O. Khatib [27] for a single obstacle case, and later extended to multiple agents [48]. The problem in this implementation [27], [48] is the existence of the spurious local minima where the agent can get stuck. In fact precisely, the local minima occur at configurations where the attractive and repulsive forces cancel each other.

Although by careful choice of potential fields the local minima may be reduced [48], there is no guarantee that they are eliminated.

2.3.4 Combination of Cell Decomposition and Potential Field Method

An instance of this method is found in [17], in which the free space is decomposed into cells and the navigation control problem for each cell is solved using a potential field built local information. The control policy is switched as the system moves from cell to cell.

Once the workspace is decomposed into cells, an adjacency graph is formed with the root node as the cell containing the goal, and a partial ordering of the cells using a graph search algorithm such as *Dijkstra's algorithm* [29] is determined. Each individual cell is then associated with a policy called *component control policy* which is so designed as to direct the robot in that specific cell along the trajectory that leads to the next adjacent cell, specified by the partial order generated by the graph search algorithm. The resulting composition of all the control policies via the partial order forms the hybrid control policy for the entire free space called the *global control policy*. The component control policy for each cell is obtained by solving the navigation problem for each cell. As the robot moves from one cell to the other cell, the control policy is switched to the one corresponding to the new cell. As the cells do not contain any obstacles, there is no problem of local minima. A similar approach for nonholonomic agents was used in [26], in which the workspace is decomposed either into triangular or rectangular torus cells depending on whether the workspace is polygonal or orthopolygonal, and a smooth feedback controller is designed for each cell to reach a particular state in the configuration space. The obstacles are assumed to be polygonal as well.

2.4 Motion planning for groups of robots

The need to control more than one agent at a time has led to the study of the problem of navigation for multiple agents. Based on the extent of information available to each agent in the workspace, the planning scheme can be classified either as centralized or decentralized.

Two types of control strategies for multiple robots based on potential fields are discussed below: the centralized, and the decentralized control schemes.

2.4.1 Centralized Navigation Scheme

In this scheme, a single navigation controller is used that has information about all the agents in the workspace. The limitation of using a centralized controller is that the required communication bandwidth and the computational load increase with the number of agents in the workspace.

A centralized architecture [49], [31], [45], [32] typically involves a single controller that depends on the configuration of all the members. Each agent has information about all the other robots in the workspace. This type of control strategy allows to generate the collision free trajectories to all the agents in the composite workspace. The gradient of the function is computed at a central location which is then communicated to all the agents. The main advantage with this scheme is that in principle, the analytical proofs of convergence can be provided in a straightforward way.

In [7], centralized planning of the trajectories of holonomic disk shaped robots was generated offline based on the potential field approach. The problem of local minima in [7] has been overcome in [8], [9] using an offline potential field based randomized path planner that used random motions to escape local minima.

The multi-agent functions constructed in [49], [31], [45] and [32] are an extension of the potential field based closed loop navigation function for a single point agent developed by Rimon and Koditschek in [28], [38]. In [31], the volume of the

agent is taken into account, with imposed holonomic constraints, using the centralized architecture scheme. However, we see that, these extensions to multiple agents is not as straightforward as it seems and the current work provides a more complete framework for generalization to multiple agents.

2.4.2 Decentralized Navigation Scheme

In the decentralized approach, each agent has information about its own position and that of the agents in an area local to its position. While centralized path planners have the disadvantage of increased complexity with increase in number of agents in the workspace, the decentralizing path planning techniques enjoy the advantage of improved memory usage and efficient communication among the agents. The constructions in [51], [21] and [20] are a direct extension to multiple agents of the single robot navigation function described in [28], similarly to how it is done in [49]; the construction in [51], [21] requires knowledge of the number of agents in the workspace, because it uses n -ary relations between agents to single out which agents are close(r) to colliding with each other at each given instance in time. In both of these approaches, there are no stationary obstacles in the environment. The idea in [28] is also extended to multiple agents using the decentralization planning scheme as can be found in [52] and [21], where the agents are assumed to have global sensing capabilities.

Formation control of multiple agents is also achieved using *structural potential functions* where the potential function for the formation is obtained from its algebraic structural constraints. In [37], the formation stabilization of n agents to a desired undirected formation is achieved using structural functions. In [30], coordinated control of multiple agents is achieved using artificial potential functions and *virtual leaders*. Virtual leader is a moving reference point that manipulates the motion of the agents in formation with its associated local potential field. The

fact that no agent is selected as a leader adds to the robustness in case of an agent failure.

In [19], the emphasis is on formation control, and the workspace includes agents as well as obstacles, with each agent having a limited communication region within which the agent can properly communicate with any other agent. In [44] and [43], the problem of formation control is treated in a scenario where agent relative position specifications are expressed through a directed graph. Unlike [52] and [21], the decentralization is limited in [44] because each agent has a copy of the global navigation function and thus needs complete group state information. In [43] on the other hand, the degree of decentralization is increased since the navigation function is decomposed into local navigation functions which are implemented and used by individual agents. In both [44] and [43], the communication capability of each agent is assumed limited.

2.4.3 Approaches based on formal languages

Some of the existing planning approaches use computer science tools like motion description languages (MDLe) and dynamic networks to navigate multiple robots. MDLe (where e stands for extended) is an extension of the framework of motion description languages introduced by Roger Brockett [11]. MDL is designed to overcome the limitations of transferring the theoretical control algorithms to software. MDLe [33] is a device independent programming language that allows the usage of hybrid controllers, and accomodates multi-system interactions and inter-agent communications.

Motion planning for multiple robots in dynamic environments using dynamic networks was described in [15] with the agents assumed to have limited sensing capabilities. This planning framework combines centralized and decentralized planning techniques for navigation. As the robots move in the workspace, the communication links among the robots keep changing and form new networks (the network area

could be the local area for which the robots' sensing capabilities exist). While the robots are in a particular network formation, they exchange the information among themselves and a trajectory is constructed using a centralized scheme. However, planning over multiple networks is decentralized. The advantages of both centralized and decentralized schemes can be utilized using this planning scheme. Though [15] is an improved version of [42] and [14], the disadvantage with these techniques is the inability to guarantee that they will provide a solution when one exists, *i.e.*, they are not complete.

2.5 Challenges

Since real world applications demand safe maneuverability of robots in uncertain workspaces, which is the general scenario in industrial areas, the navigation of robots in dynamic environments certainly requires efficient real time online planning to reduce the uncertainties which can be made possible through potential field based navigation functions. The need to have a *complete* planner is also important, and establishing theoretical correctness is the first step towards demonstration in practical applications. As we have seen earlier the potential field based navigation functions for a single agent enjoy the advantage of global optimal convergence and hence are complete.

The existing techniques for multi-robot systems, as it may be expected, are the above generalizations of the single-robot navigation function to the multiple agents which follow the analysis steps of the original construction in [28]. The assumptions made about the topology of the environment and the robots can be directly related, for the most part, to the assumptions appearing in the single-agent case of [52] and its first multi-agent generalization in [49]. It turns out, however, that some of these assumptions unavoidably break down in the multi-agent case. One of these critical assumptions that may have been overlooked is the one that requires that obstacles are isolated. In the single-robot case where

this assumption was originally imposed, the statement implies that as the system approaches a collision configuration (with an obstacle), there is a single obstacle the distance to which continuously decreases, whereas distances to other obstacles remain bounded above zero. In the multi-robot case, collisions can occur between multiple robots, and since all robots can move, it is not clear why only two robots can collide with each other at any given time; in fact, it is conceivable that all robots collapse on each other simultaneously, and the *n-ary* relations of [52], [21] are a testimony to that.

In view of the above, we find that, in the existing work it is not entirely clear how convergence proofs for single agents generalize to multiple agents in a mathematically rigorous way. We, therefore, study the potential field based navigation functions, and provide a new framework for the centralized scheme of planning for multi robot navigation.

Chapter 3

A CLOSER LOOK AT EXISTING METHODS

In this chapter we will have a closer look at the existing navigation function based methods to steer multiple robots. We reveal some of the open issues associated with these methods which we highlight by the end of this chapter.

Informal definitions of few mathematical terms are given below. The intention here is to provide some intuitive explanation to facilitate further reading, and not to give a precise definition. Mathematical definitions can be found in the Appendix.

1. **Compact Set:** A subset of a metric space is said to be *compact* if it is both *closed* and *bounded*.
2. **Connected Space:** A topological space is said to be connected, if any two points in that space can be connected by a path, with the path itself being a proper subset of that topological space.
3. **Analytic function:** A function is said to be analytic if at every point on its domain there exists a Taylor series that converges to that point. Analytic functions are infinitely differentiable.
4. **Manifold:** A manifold is a metric space in which the neighborhood of each point is topologically equivalent to a Euclidean space of a particular dimension, say n , known as the dimension of the manifold.

5. **Manifold with boundary:** A closed manifold is a **manifold with boundary**. It consists of both the interior points and the points on its boundary.
6. **Critical Point:** A critical point of a function is that point at which the function's gradient becomes zero.
7. **Hessian of a function:** The Hessian of a multi-variable function is a matrix of second partial derivatives of the function.
8. **Non-degenerate critical point:** A critical point is non-degenerate if the function's Hessian at that point is full-rank.

3.1 Single agent navigation function construction

The major drawback of the potential field technique, the problem of local minima, is overcome by Rimon and Koditschek in [28] by a new construction that replaces the superposition of attractive and repulsive fields for the goal and obstacles respectively, with a gradient field of a single potential function, specially designed so that its critical points have particular properties.

In its original “model” form, the *navigation function* [38] was defined for a single robot agent, with trivial, single integrator kinematics

$$\dot{q} = u \text{ ,}$$

moving in a workspace given by

$$\mathcal{W} \triangleq \{q \in \mathbb{R}^n : \|q\|^2 \leq \rho_0^2\}.$$

The workspace \mathcal{W} is populated by M spherical obstacles given by

$$\mathcal{O}_i \triangleq \{q \in \mathbb{R}^n : \|q - c_i\|^2 < \rho_i^2\} \text{ ,}$$

where each \mathcal{O}_i is an open set. Here, c_i and ρ_i are the center and radius of obstacle i , respectively. The boundary of \mathcal{W} is referred to as the zeroth obstacle, centered at the origin. Then the free workspace for the robot is the set

$$\mathcal{F} \triangleq \mathcal{W} \setminus \bigcup_{i=1}^M \mathcal{O}_i .$$

This workspace \mathcal{F} is said to be *valid* if the closures of all \mathcal{O}_i are in the interior of \mathcal{F} , and that none of them intersect:

$$\|c_i - c_j\| > \rho_i + \rho_j \tag{3.1}$$

Definition 4 (Navigation Function [38]). *Let $\mathcal{F} \subset \mathbb{R}^n$ be a compact connected analytic manifold with boundary. A map $\varphi : \mathcal{F} \rightarrow [0, 1]$ is a navigation function if φ is*

1. *Analytic on $\mathcal{F} - \mathcal{M}$, where \mathcal{M} is a set of measure zero.*
2. *Polar on \mathcal{F} , i.e., φ has a single minimum on its domain at q_d .*
3. *Morse on \mathcal{F} , i.e., all the its critical points are non-degenerate.*
4. *Admissible on \mathcal{F} , i.e., φ attains a uniform maximal value on its boundary.*

The first requirement of φ being analytic is due to the intention of the authors of [28], to make the controller commands to the robot analytically simple. If φ is analytic, it can be expressed in closed form as a single mathematical function, which makes the computation of its gradient straightforward. This avoids patching together gradients of multiple functions. However, if φ is analytic, it is a C^∞ function, which is too much in terms of ensuring the desired properties for the function, a C^2 differentiability for φ would suffice.

Having φ as a polar function means that it has have a unique minimum at the destination configuration q_d on its domain \mathcal{F} . Though it is highly likely that φ

can have local minima, we see that with appropriate tuning of a parameter κ , they can be moved close to the obstacles and turned into saddles.

Admissible functions, as defined earlier, have a uniform maximum value on their boundary. An admissible φ ensures that the robot does not slide along the boundary of the free space. With a uniform value on the boundary, the gradient of the function on that boundary would always point into the interior of the free space.

For φ to be Morse [35], all its critical points have to be non-degenerate. A non-degenerate critical point is either a local minimum, or a local maximum or a saddle point. The navigation function φ , once tuned appropriately, will exhibit a single minimum q_d , and have a number of possible saddles in the free space. If the conditions on the robot agent are set away from the set of measure zero \mathcal{M} which is the region of attraction of the saddle points, then all integral lines of the field will converge to q_d .

3.2 The goal function

The goal function $\gamma : \mathcal{F} \rightarrow [0, \infty)$ is a metric that measures how close the robot is to the destination configuration. It is defined using the Euclidean norm as

$$\gamma(q) \triangleq \gamma_d^\kappa(q) \text{ ,} \tag{3.2}$$

with

$$\gamma_d \triangleq \|q - q_d\|^2 \text{ ,} \tag{3.3}$$

where q is the robot configuration at which the the goal function is calculated, q_d is the destination configuration, and κ is the navigation function's tuning parameter which regulates the shape of the graph of the goal function, and consequently that of φ .

3.3 The obstacle function

The obstacles \mathcal{O}_i are modeled initially as sphere-shaped objects. In a realistic workspace, not all obstacles are sphere-shaped, however, it was shown in [38] that a star-shaped obstacle can be transformed into a sphere-shaped obstacle through a diffeomorphic mapping. The proximity of the agent to obstacle \mathcal{O}_i is captured using the Euclidean norm in a positive semi-definite obstacle function β_i . Each β_i thus corresponds to some isolated obstacle \mathcal{O}_i .

An individual obstacle function is defined as,

$$\beta_i(q) \triangleq \|q - c_i\|^2 - \rho_i^2 , \quad (3.4)$$

where ρ_i is the radius of the obstacle i , and c_i its center with i ranging from 1 to M .

The boundary of the workspace is considered as the “zeroth” obstacle and defined as

$$\beta_0(q) \triangleq \rho_0^2 - \|q\|^2 ,$$

where ρ_0 is the radius of the workspace boundary.

A measure of proximity of the robot to the whole collection of obstacles can be constructed by multiplying all the individual obstacle functions β_i . The obstacle function for the workspace is constructed as the product of individual obstacle functions is

$$\beta(q) \triangleq \prod_{i=0}^M \beta_i(q) \quad (3.5)$$

where i ranges from 0 to M . When robot is close to the i^{th} obstacle the corresponding obstacle function β_i approaches zero and hence the function β vanishes.

A first attempt to constructing a navigation function is as

$$\varphi = \frac{\gamma}{\beta} . \quad (3.6)$$

However, the above function fails to have the properties defined in definition [28] and hence is not a navigation function in a strict sense. It is polar, Morse

everywhere except at q_d , and is admissible by attaining ∞ on the boundary. Function (3.6) can be transformed into another function that has the desired properties by the series of transformations described in Section 3.3.1.

3.3.1 Analytical switches

A diffeomorphism is a differentiable bijective map between manifolds. These functions preserve the properties of the navigation function [38]. Different diffeomorphisms are also used to transform the star-shaped obstacles into spherical obstacles. Under these obstacle transformations the navigation properties of the workspace do not change, as shown in [38]. The first diffeomorphism used here is to reduce the range of (3.6) to make the function bounded. The following function $\sigma : [0, \infty) \rightarrow [0, 1]$ is a diffeomorphism whose range is constrained to $[0, 1]$.

$$\sigma = \frac{x}{1+x} \quad (3.7)$$

The function given by (3.6) is polar, is admissible, and is Morse almost everywhere except at the destination q_d . When this function is put through the transformation σ given by (3.7) we get,

$$\varphi = \frac{\gamma}{\gamma + \beta}$$

which is thus polar, admissible, analytic and is non-degenerate almost everywhere except at the destination. To make the above function non-degenerate even at the destination, the below transformation $\sigma_d : [0, 1] \rightarrow [0, 1]$, is used.

$$\sigma_d(x) = (x)^{1/\kappa}, \kappa \in \mathbb{N} . \quad (3.8)$$

The result of this transformation is the following function with the properties given in Definition 4

$$\varphi = \frac{\gamma_d}{[\gamma_d^\kappa + \beta]^{1/\kappa}} . \quad (3.9)$$

3.4 Decomposition of the workspace \mathcal{F}

To describe the behavior of φ , the workspace \mathcal{F} is first decomposed into different regions. Let ϵ be a small positive parameter which defines the neighborhood of the obstacles and the workspace boundary.

The destination configuration, $\{q_d\}$, is the single global minimum of the navigation function in the bounded workspace \mathcal{F} .

Around each obstacle, \mathcal{O}_i , a region is constructed as a function of ϵ . Represented by $\mathcal{B}_i(\epsilon)$, it is defined as

$$\mathcal{B}_i(\epsilon) \triangleq \{q \in \mathcal{W} : 0 < \beta_i < \epsilon\} .$$

Since the boundary of the workspace is considered as the “zeroth” obstacle, $\mathcal{B}_0(\epsilon)$ represents the neighborhood region close to the workspace outer boundary.

The boundary of the free space, where the navigation function vanishes, is denoted by $\delta\mathcal{F}$.

The union of $\mathcal{B}_i(\epsilon)$ regions of all obstacles \mathcal{O}_i is denoted $\mathcal{F}_0(\epsilon)$, and defined as

$$\mathcal{F}_0(\epsilon) \triangleq \bigcup_{i=1}^M \mathcal{B}_i \setminus \{q_d\} .$$

The region “close” to the workspace boundary is the region $\mathcal{B}_0(\epsilon)$ excluding the destination configuration q_d and any $\mathcal{B}_i(\epsilon)$ regions of other obstacles that might overlap with the $\mathcal{B}_0(\epsilon)$. It is defined as

$$\mathcal{F}_1(\epsilon) \triangleq \mathcal{B}_0 \setminus (\{q_d\} \cup \mathcal{F}_0) .$$

The region away from the obstacles includes the entire workspace \mathcal{W} excluding the destination configuration, the regions \mathcal{F}_0 , \mathcal{F}_1 and $\delta\mathcal{F}$. It is defined as

$$\mathcal{F}_2(\epsilon) \triangleq \mathcal{F} \setminus (\{q_d\} \cup \delta\mathcal{F} \cup \mathcal{F}_0 \cup \mathcal{F}_1) .$$

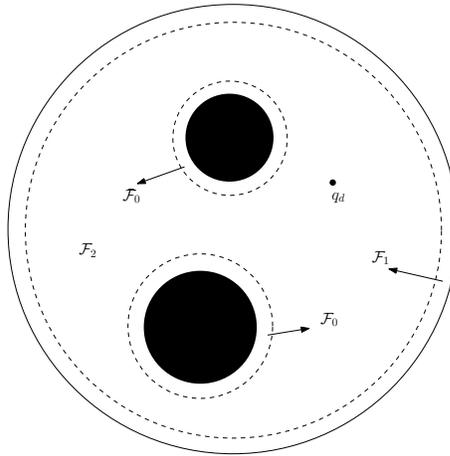


Figure 3.1: A figurative description of the decomposition of the workspace \mathcal{F} into different regions. \mathcal{F}_0 represents the neighborhood of the obstacles. \mathcal{F}_1 represents the neighborhood of the boundary of workspace $\partial\mathcal{F}$. q_d denotes the destination configuration in the free workspace \mathcal{F}_2 .

3.5 Why φ is a navigation function

The sequence of propositions that establish the proof of correctness for φ when the *workspace is valid* are stated informally as follows:

1. The destination q_d is a non-degenerate local minimum of φ
2. The critical points of φ are in the interior of the free space
3. For every ϵ one can choose a κ so that $\frac{\gamma}{\beta}$ has no critical points away from obstacles
4. There exists a lower bound for ϵ , below which the critical points of $\frac{\gamma}{\beta}$ close to the obstacle boundary are not local minima
5. There is another lower bound for ϵ , below which there are no critical points close to the workspace boundary

6. There is a last lower bound for ϵ , below which whatever critical points besides the destination are inside the free space, are non-degenerate (and therefore, they have to be saddles).

The detailed proofs for these propositions can be found in [28].

3.6 Attempts at extending to multiple agents

To extend the single-agent construction to multiple agents the number of agents is increased from one to N . The notation is changed slightly to denote this fact, and for simplicity we assume that there are no stationary obstacles in the agents' environment. Rather, the agents may run into each other, and therefore each one of them poses an obstacle to all others. The dynamics of each agent is:

$$\dot{p}_i = u_i, \quad p_i \in \mathbb{R}^n, \quad i = 1, \dots, N \quad (3.10)$$

where p_i and u_i are the position and control input of agent i , respectively. Let p denote the stack vector of all p_i , and define \mathcal{P} as the set of p for which $\|p_i - p_j\| \geq \rho$, for all $i, j = 1, \dots, N$. Set \mathcal{P} plays the role of \mathcal{F} here.

A straightforward approach followed in existing literature is to start with (3.9), define a goal function γ_d and the obstacle function β in a similar way as in (3.4)-(3.5) and then set

$$\varphi = \frac{\gamma}{(\gamma^\kappa + \beta)^{1/\kappa}}. \quad (3.11)$$

These choices have advantages and disadvantages: on one hand, a γ_d defined as in (3.2)-(3.3) requires each robot to achieve a pre-specified position in the workspace and thus does not allow the formation to “float” freely in space; on the other hand it ensures that the destination point will eventually turn out to be a non-degenerate critical point in \mathcal{P} .

In the original construction of the navigation function [28], the obstacle function β was defined as the product of individual obstacle functions β_i . When the agent

approaches any obstacle \mathcal{O}_i , β_i would tend to zero. When the same exact methodology is extended to scenarios with multiple agents, it can happen that any number of agents, not limited to two, can come close enough to each other in which case more than one β_i would tend to zero. In such cases with sufficient number of β_i vanishing at the same time one can have $\nabla^2\beta$ vanishing too. This makes the task of proving the claim 4 in the list of the previous section, impossible (at least using the known approach of [28]).

The approach used in [28] to establish the Morse properties of the navigation function makes use of the following lemma.

Lemma 1. ([28, Lemma 3.1]) *Let $\mathbb{R}^n = \mathcal{P} \oplus \mathcal{N}$ and let the symmetric matrix $Q \in \mathbb{R}^{n \times n}$ define a quadratic form on \mathbb{R}^n .*

$$\xi(v) \triangleq v^T Q v$$

If $\xi|_{\mathcal{P}}$ (the restriction of ξ in \mathcal{P}) is positive definite and $\xi|_{\mathcal{N}}$ is negative definite, then Q is non-singular and

$$\text{index}(Q) = \dim(\mathcal{N}) .$$

The set of all critical points of φ is denoted \mathcal{C}_φ . Any critical point of φ in \mathcal{F}_0 is naturally in $\mathcal{C}_\varphi \cap \mathcal{F}_0$. The tangent space at $q \in \mathcal{C}_\varphi \cap \mathcal{F}_0$ is decomposed into two orthogonal components \mathcal{P} and \mathcal{Q} . The idea is to express the quadratic form of the Hessian at each critical point and a vector in both \mathcal{P} and \mathcal{Q} and show that they are positive-definite and negative-definite respectively.

At the critical point,

$$\nabla^2 \left(\frac{\gamma}{\beta} \right) \propto 2 \frac{\|\nabla\beta\|}{\|\nabla\gamma\|} I - \nabla^2\beta . \quad (3.12)$$

Let us see through an example why a direct extension might fail. While computing $\nabla^2\beta$ for the case of multiple agents, it is possible that more than one of

β_i might tend to zero. For an example with four agents, the obstacle function would be $\beta(x) = \beta_1(x) \cdot \beta_2(x) \cdot \beta_3(x) \cdot \beta_4(x)$.

The gradient of β would be

$$\begin{aligned} \nabla\beta(x) &= \beta_1'(x) \cdot \beta_2(x) \cdot \beta_3(x) \cdot \beta_4(x) + \beta_1(x) \cdot \beta_2'(x) \cdot \beta_3(x) \cdot \beta_4(x) \\ &\quad + \beta_1(x) \cdot \beta_2(x) \cdot \beta_3'(x) \cdot \beta_4(x) + \beta_1(x) \cdot \beta_2(x) \cdot \beta_3(x) \cdot \beta_4'(x) \end{aligned}$$

The Hessian in this case would be

$$\begin{aligned} \nabla^2\beta(x) &= 2\beta_3(x) \cdot \beta_4(x) \cdot (\beta_1)'(x) \cdot (\beta_2)'(x) + 2\beta_2(x) \cdot \beta_4(x) \cdot (\beta_1)'(x) \cdot (\beta_3)'(x) \\ &\quad + 2\beta_1(x) \cdot \beta_4(x) \cdot (\beta_2)'(x) \cdot (\beta_3)'(x) + 2\beta_2(x) \cdot \beta_3(x) \cdot (\beta_1)'(x) \cdot (\beta_4)'(x) \\ &\quad + 2\beta_1(x) \cdot \beta_3(x) \cdot (\beta_2)'(x) \cdot (\beta_4)'(x) + 2\beta_1(x) \cdot \beta_2(x) \cdot (\beta_3)'(x) \cdot (\beta_4)'(x) \\ &\quad + \beta_2(x) \cdot \beta_3(x) \cdot \beta_4(x) \cdot (\beta_1)''(x) + \beta_1(x) \cdot \beta_3(x) \cdot \beta_4(x) \cdot (\beta_2)''(x) + \\ &\quad \beta_1(x) \cdot \beta_2(x) \cdot \beta_4(x) \cdot (\beta_3)''(x) + \beta_1(x) \cdot \beta_2(x) \cdot \beta_3(x) \cdot (\beta_4)''(x) \end{aligned} \quad (3.13)$$

With any combination of three of $\beta_1(x), \beta_2(x), \beta_3(x)$ and $\beta_4(x)$ approaching zero, the Hessian $\nabla^2\beta$ in (3.13) approaches zero. If $\nabla^2\beta \rightarrow 0$ the Hessian of the navigation function at a critical point (see (3.12)) would become positive and hence a negative eigenvalue for $\nabla^2\left(\frac{\gamma}{\beta}\right)$ can never be found. It is thus unclear how to prove the Proposition 4 for multiple agents.

Having said this, it is not necessarily the case that a potential function built in this way may fail to stabilize a multi-agent formation; in fact, reported numerical results point to the opposite. This could be because potential problems may arise only in extreme situations that may not be predicted beforehand. It could also be the case, for example, that a critical point in question is not a local minimum but rather just a degenerate critical point. Such critical points present problems mainly when the system is initialized in their zero measure attraction region, or when one applies diffeomorphic transformations to the configuration space.

3.7 Challenges

Having identified the limitation of the choice of the obstacle function as a product of individual obstacle functions, we conclude that the extension of the navigation function methodology to multiple agents remains a challenge.

Using the existing approach, establishing the Morse properties of the navigation function close to the obstacles has not been formally achieved. We believe that this challenge can be overcome by using an alternate construction for the navigation function for which the eigenvalues of the Hessian can be decomposed into strictly positive and negative values using the same approach.

We present such an alternate form of the obstacle function β , in the form of a function of the distance between the two nearest agents in the workspace. For the agents not to have an influence over each other when they are far away, we construct β in a way that it approaches a constant value beyond certain distance between the nearest agents. This ensures that the interaction between the agents remains local.

In certain cases where the minimal distance could be between more than two agents, it turns out that the obstacle function is non-differentiable. We hence present concepts from nonsmooth analysis in the next chapter, which are later utilized in the proof of correctness for the new multi-agent navigation function construction.

Chapter 4

NONSMOOTH ANALYSIS AND THE DISTANCE FUNCTION

This chapter deals with the analysis of nonsmooth functions. These functions represent the class of functions which are not differentiable everywhere. To define the solutions of a gradient flow at points where the gradient switches we use the concepts from the nonsmooth analysis. As we reach the end of this chapter we see some of the differential properties of such functions and the conditions under which these properties can be applied. The results presented here are utilized later for establishing the proof of correctness of φ . Most of the results in this section are from [39].

4.1 Generalized directional derivative

At a point where a function does not have (even a directional) derivative a generalized derivative can be defined.

Definition 5 (Generalized directional derivative). *The Clarke generalized derivative of $f(x)$ at x in the direction v is defined as*

$$f^o(x; v) = \limsup_{y \rightarrow x} \sup_{h \downarrow 0} \frac{f(y + hv) - f(x)}{h} \quad (4.1)$$

For the generalized derivative, since only the upper limit is considered, the limit need not exist at x . We now define the right directional derivative which is used in the definition of a regular function.

Definition 6 (Right directional derivative). *The right directional derivative of $f : \mathbb{R}^d \rightarrow \mathbb{R}$ at x in the direction of $v \in \mathbb{R}^d$ is defined as*

$$f'(x; v) = \lim_{h \rightarrow 0^+} \frac{f(x + hv) - f(x)}{h},$$

4.2 Generalized gradient

Definition 7. *The Clarke generalized gradient of $f(x)$ at $x \in X^*$ is*

$$\partial f(x) = \{\zeta \in X^* \mid f^\circ(x; v) \geq \langle \zeta, v \rangle\} \quad (4.2)$$

where X^* is the dual space of continuous linear functionals on X , $f^\circ(x; v)$ generalized derivative of $f(x)$ at x in the direction v and $\langle \zeta, v \rangle$ denotes the value of functional ζ at v .

In fact, when X is finite dimensional, and if $\Omega_f \subset X$ is the set of points where f is not differentiable, we can write

$$\partial f(x) = \text{co}\left\{\lim_{i \rightarrow \infty} \nabla f(x_i) : x_i \rightarrow x, x_i \notin \mathcal{M} \cup \Omega_f\right\} \quad (4.3)$$

where co stands for the convex hull, \mathcal{M} can be any set of measure zero, and x_i any sequence converging to x .

The generalized gradient is thus the convex hull of all the points of the form $\lim \nabla f(x_i)$, where $\{x_i\}$ is a sequence converging to x avoiding $\mathcal{M} \cup \Omega_f$.

4.3 Generalized gradient-pointwise minima

In this section, we deal with the generalized gradient for a function that attains pointwise minima. An existing result on pointwise maxima is quoted and is utilized to prove the result on pointwise minima.

Definition 8 (Regular Function). *A function is said to be regular when its right directional derivative and its generalized derivative are equal i.e.,*

$$f'(x; v) = f^\circ(x; v)$$

We present here a result related to the generalized gradient of a function that is defined as a pointwise maximum ([16], Proposition 2.3.12).

Suppose $\{f_i\}$ is a finite collection of functions ($i = 1, 2, \dots, n$) each of which is Lipschitz near x . The function f defined by

$$f(x') = \max\{f_i(x') : i = 1, 2, \dots, n\}$$

is also Lipschitz at x . For any x' we let $I(x')$ denote the set of indices for which $f_i(x') = f(x')$ (i.e., the indices at which the maximum defining f is attained). The proof for the following proposition can be found in [16].

Proposition 1 (Pointwise maxima).

$$\partial f(x) \subset \text{co}\{\partial f_i(x) : i \in I(x)\} ,$$

and if f_i is regular at x for each i in $I(x)$, then equality holds and f is regular at x .

Using the above proposition we compute the generalized gradient for the case of pointwise minima too. The following proposition was not found in the literature in the form stated below, so a proof for it is provided. Stronger versions of this proposition can be shown for the case where the functions f_i are convex or regular, similar to the stronger version (i.e., the case of equality) of pointwise maxima as seen in the previous proposition.

Proposition 2 (Pointwise minima).

$$\partial \min_{i=1, \dots, n} f_i(x) \subset \text{co}\{\partial f_i(x) : i = 1, \dots, n\}$$

Proof.

$$\begin{aligned}
\partial \min_{i=1,\dots,n} f_i(x) &= \partial(-\max\{-f_i(x)\}) \\
&= -\partial(\max\{-f_i(x)\}) \\
&\stackrel{[Proposition 2]}{\subset} -\text{co}\{\partial(-f_i(x)) : 1 \leq i \leq n\} \\
&= -\text{co}\{-\partial(f_i(x)) : 1 \leq i \leq n\} \\
&= \text{co}\{\partial(f_i(x)) : 1 \leq i \leq n\}
\end{aligned} \tag{4.4}$$

□

4.4 Filippov Solutions

Filippov has developed a technique for finding solutions to differential equations with discontinuous right hand sides [1].

At a point $x \in \mathbb{R}^d$, the vector field $\mathcal{X}(x(t))$ is evaluated at $B(x, \delta)$, an open ball centered at x with radius $\delta > 0$. The effect of δ approaching zero is examined by evaluating \mathcal{X} for smaller and smaller δ . The notion of Filippov solution is based on the *differential inclusion* of the nonsmooth equation.

Definition 9 (Set-Valued Map [18]). *A set-valued map is a map that assigns sets to points. The set-valued maps are of the form $\mathcal{F} : [0, \infty) \times \mathbb{R}^d \rightarrow \mathcal{B}(\mathbb{R}^d)$ where $\mathcal{B}(\mathbb{R}^d)$ denotes the collection of all subsets of \mathbb{R}^d . The map \mathcal{F} assigns to each point $(t, x) \in [0, \infty) \times \mathbb{R}^d$ the set $\mathcal{F}(t, x) \subseteq \mathbb{R}^d$.*

Differential inclusion and the notion of Filippov solution are defined as follows.

Definition 10 (Filippov solution, Differential Inclusion [1]). *Let the vector-valued function $X(t, x)$ be defined almost everywhere and measurable in the domain G of*

the (t, x) -space ($x \in \mathbb{R}^n$) and let there exist, for each bounded closed domain $D \subset G$, an almost everywhere finite solution $m(t)$ such that

$$|X(t, x)| \leq m(t)$$

almost everywhere in D .

Let $F(t, x)$ be the smallest closed convex set containing all limit values of the vector function $X(t, x')$, where tending to x, x' spans almost the whole neighborhood (that is, except for a set of measure zero) of the point x , that is,

$$F(t, x) = \bigcap_{\delta > 0} \bigcap_{\mu(S)=0} \overline{\text{co}}\{X(t, x^\delta) \setminus S\} .$$

Here $\overline{\text{co}}$ implies convex closure; intersection is taken over all sets S of measure zero and over all $\delta > 0$, and, F is the Filippov set-valued map.

Consider an equation or a system in vector notation

$$\dot{x} = X(t, x) \tag{4.5}$$

with a piecewise continuous function X in a domain G , $x \in \mathbb{R}^n$, $\dot{x} = \frac{dx}{dt}$; M is a set (of measure zero) of points of discontinuity of the function X . A solution of equation (4.5) is called a solution of the differential inclusion

$$\dot{x} \in F(t, x) , \tag{4.6}$$

that is, an absolutely continuous vector-valued function $x(t)$ defined on an interval or on a segment I for which $\dot{x}(t) \in F(t, x(t))$ almost everywhere on I .

Let for each point $(t, x) \in G$ the set $F(t, x)$ be the smallest closed convex set containing all the limit values of the vector-valued function $X(t, x^*)$ for $(t, x^*) \notin M$, $x^* \rightarrow x$, $t = \text{const}$. A solution of equation (4.5) is a Filippov solution of the inclusion (4.6).

A point x_e is an equilibrium of the differential inclusion if $0 \in \mathcal{F}(t, x_e)$ for all $t \in [0, \infty)$. The possible evolutions of the system is defined using a set-valued map $\mathcal{F}(t, x) : [0, \infty) \times \mathbb{R}^d \rightarrow \mathcal{B}(\mathbb{R}^d)$.

Definition 11 (Maximal solution [39]). Consider the differential inclusion given by

$$\dot{x} \in X(x(t)), \quad x \in \mathcal{G}, \quad x(t_0) = x_0 \quad (4.7)$$

A solution x of this inclusion is said to be maximal if it does not have a proper right extension i.e. there is no solution defined on an interval $\tilde{\mathcal{G}}$ which properly contains the interval \mathcal{G} .

In other words, there is no solution x in $\tilde{\mathcal{G}}$ such that $\mathcal{G} \subset \tilde{\mathcal{G}}$.

Definition 12 (Precompact solution [39]). A solution $x(t)$ of (4.7) is said to be precompact, if it is maximal and the closure $\text{cl}(x([t_0, \omega]))$ of $x(t)$ is a compact subset of \mathcal{G} .

Definition 13 (Limit set [39]). Let $x(t)$ be a maximal solution of (4.7). A point $\bar{x} \in \mathbb{R}^N$ is an ω -limit point of $x(t)$ if there exists an increasing sequence $t_n \subset [0, \omega)$ such that $t_n \rightarrow \omega$ with $n \rightarrow \infty$ implies that $x(t_n) \rightarrow \bar{x}$. The set $\Omega(x)$ of all limit points of the solution $x(t)$ is the ω -limit set of $x(t)$.

Definition 14 (Weak invariance [39]). With respect to (4.7), a set $\mathcal{S} \subset \mathbb{R}^N$ is weakly invariant if for each $x(0) \in \mathcal{S} \cap \mathcal{G}$, there exists at least one maximal solution of $x(t)$ of (4.7) with $\omega = \infty$ and with $x([0, \omega)) \in \mathcal{S}$. We say that \mathcal{S} is strongly invariant if every trajectory emanating from $x(0)$ remains in \mathcal{S} with $\omega \rightarrow \infty$.

The following proposition is the nonsmooth equivalent of the known result that states that differentiable solutions bounded in a compact set have a nonempty attractive invariant set.

Proposition 3 ([39]). If x is a precompact solution of (4.7), then $\Omega(x)$ is a nonempty, compact, connected subset of \mathcal{G} . Moreover $\Omega(x)$ is the smallest closed set approached by x and is weakly invariant.

We next present Ryan’s version of the invariance principle [39]. Compared to other results available in literature [5], Ryan’s version differs for not imposing regularity assumptions on the Lyapunov-like function.

Theorem 1 ([39]). *Let $V : \mathcal{G} \rightarrow \mathbb{R}$ be locally Lipschitz. Define*

$$u : \mathcal{G} \rightarrow \mathbb{R}, z \rightarrow u(z) := \max\{V^o(z, v) | v \in X(z)\} ,$$

Suppose that $U \subset \mathcal{G}$ is non-empty and that $u(z) \leq 0$ for all $z \in U$,

If x is a precompact solution of (4.7) with trajectory in U , then, for some constant $c \in V(\text{cl}(U) \cap \mathcal{G})$, x approaches the largest weakly invariant set in $\Sigma \cap V^{-1}(c)$, where

$$\Sigma = \{z \in \text{cl}(U) \cap \mathcal{G} | u(z) \geq 0\}$$

4.5 The distance function and its properties

The obstacle function in the new construction ((5.4)) is modeled as a function of the minimum of the distances among all the robots. In the case of multiple agents, the function is nondifferentiable, and its nonsmooth properties affect those of the gradient of the navigation function.

Definition 15 (The distance function). *The Euclidean distance function d_c of a point v to set $C \subset \mathbb{R}^n$ is defined as*

$$d_c(v) = \inf\{\|v - c\| : c \in C\}$$

4.6 The distance function in the new multi-agent navigation function

The distance function, in the present case, is the minimum among all the different inter-agent distances, q_{ij} , *i.e.*,

$$d(q) \triangleq \min_{ij} \{\|q_{ij}\|\} , \tag{4.8}$$

The distance function d of a point x to a set Ω is typically defined as the minimum norm of the difference between the point and any other point in the set.

$$d(x) \triangleq \min_{z \in \Omega} \|x - z\| .$$

Comparing to (4.8), x is identified with q and the set Ω is the manifold where any of the components of q becomes zero.

With the exception of trivial cases, q cannot be continuously visualized in three dimensions. The easiest case that can be reasonably depicted is that of three planar agents: the formation configuration can be uniquely described in terms of the horizontal and vertical relative position of agents 1 and 2, x_{12} and y_{12} , respectively, and the horizontal and vertical position of agents 1 and 3, x_{13} and y_{13} , respectively. In this case, q is (still) four-dimensional. But we can attempt to visualize the zero level sets of (4.8) using three-dimensional slices of this four-dimensional space (see Figure 4.1).

The distance-to-collision function is generally nonsmooth, not only at the origin, but anywhere agents that were originally the closest are not anymore (the pair of closest neighbors changes). Several interesting properties of the distance function and its (generalized) derivative come handy.

We define the finite sets

$$\begin{aligned} \Omega &\triangleq \{p_i \in \mathbb{R}^n \mid 1 \leq i \leq N\}, \\ \Omega_i &\triangleq \{p_j \in \Omega \mid j \neq i\} . \end{aligned}$$

Set Ω includes all agent position vectors, and Ω_i just excludes the position of agent i . These two sets allow us to express the distance function we introduced in (4.8), in the form,

$$d(q) \triangleq \min_{\substack{i, j \in \{1, \dots, N\} \\ i \neq j}} \|q_{ij}\| ,$$

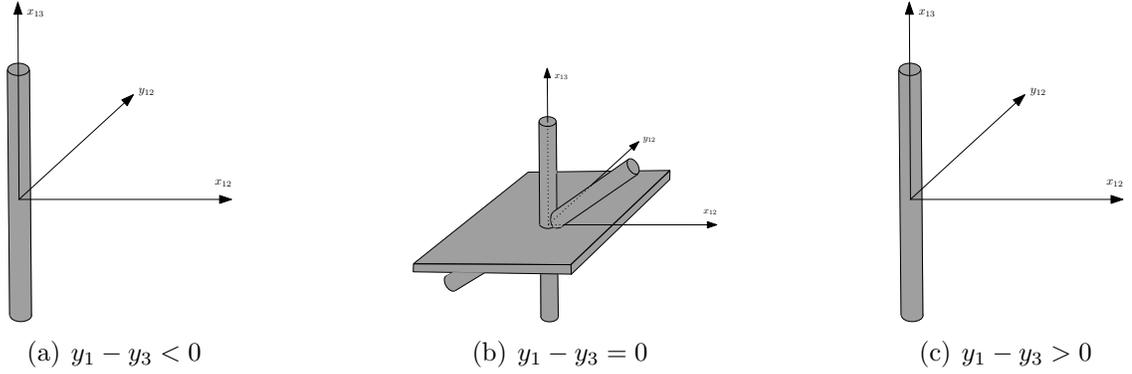


Figure 4.1: Three 3-D slices of an inter-agent distance function defined in a four-dimensional space. The collision configurations between agents 1 and 2 are marked by the cylinder that contains the (hyper)line $x_{12} = y_{12} = 0$. In the three-dimensional slices where the dimension y_{13} is not pictured, the collision configurations between agents 1 and 3 are shown as the “thick” hyperplane passing through the origin on the $y_1 - y_3 = 0$ slice. Note the diagonal cylinder with axis on the x_{13} - x_{12} plane: this represents collisions between agents 2 and 3 (although x_{23} and y_{23} are not mapped). This diagonal collision region expresses the fact that when $q_{12} = q_{13}$, agents 2 and 3 overlap; at the slice where $y_{13} = 0$, therefore, and on the plane where $y_{12} = 0 = y_{13}$, the diagonal line $x_{12} = x_{13}$ maps configurations where all three agents have the same y coordinate, and agent 2 is on top of agent 3. These three graphs illustrate that pairwise obstacle functions (i.e., collision between 1 and 2, or collision between 1 and 3) define regions in the relative position space which are not isolated, and irrespectively of the agents’ volume the origin of this space will always be a point common to all regions.

in terms of the distance d_i between a point $p_i \in \Omega$ and the rest of its groupmates which form the set Ω_i ,

$$d_i(p_i) \triangleq \min_{z \in \Omega_i} \|p_i - z\| = \min_{\substack{j=1, \dots, N \\ i \neq j}} \|q_{ij}\| \quad (4.9)$$

as follows:

$$d(q) = \min_{1 \leq i \leq N} d_i(p_i) = \min_{i=1, \dots, N} \min_{\substack{j=1, \dots, N \\ i \neq j}} \|q_{ij}\| . \quad (4.10)$$

With reference to the point-to-set distance d_i , we define the set

$$W(p_i) = \{z \in \Omega_i \mid \|p_i - z\| = d_i(p_i)\} ,$$

which identifies the nearest neighbor(s) of agent i . The indices of the nearest neighbors (the agents equidistant from i with the smallest distance from i) are contained in the sets

$$I_i = \{j \in \{1, \dots, N\} : p_j \in W(p_i)\} .$$

Existing literature [22, Theorem 1] allows us to state the following fact about the distance d_i , defined in (4.9) in terms of the Euclidean norm:

Corollary 1. *For $p_i \neq p_j$, for all $j \in \{1, \dots, N\} \setminus \{i\}$, (i.e., when $q_{ij} \neq 0$),*

$$\partial d_i(p_i) = \text{co} \{ \partial \|q_{ij}\| \mid j \in I_i \} .$$

Note that away from $q_{ij} = 0$, $\partial \|q_{ij}\| = \nabla \|q_{ij}\| = \frac{p_j - p_i}{\|q_{ij}\|}$, i.e., the unit vector that points away from p_i and toward p_j .

As a result of proposition 2, we can write

$$\partial d(q) \subset \text{co} \{ \partial \|q_{ij}\| : i, j \in \{1, \dots, N\}, j \in I_i \} ,$$

where the derivatives are taken with respect to q_{ij} . Away from points where there exists a pair $i, j \in \{1, \dots, N\}$ such that $q_{ij} = 0$, $\partial \|q_{ij}\|$ is a singleton (recall that gradients are taken with respect to q_{ij} , so in each case the derivative is that of the distance of a vector from the origin) and therefore, given (4.3),

$$\partial d(q) \subset \text{co} \left\{ \lim_{q' \rightarrow q} \nabla_q \|q_{ij}\| : i \in \{1, \dots, N\}, q_{ij} \neq 0, j \in I_i \right\} .$$

It may not be obvious, but these convex hulls never contain the zero vector away from $q_{ij} = 0$. This may become clear with the following two-dimensional example.

4.7 Example: generalized gradient of pointwise minima

Let $f(x, y) = \min\{|x|, |y|\}$, in which case we have $f_1(x, y) = |x|$, and $f_2(x, y) = |y|$. The graph of $f(x, y)$ is shown in Figure 4.2(a)

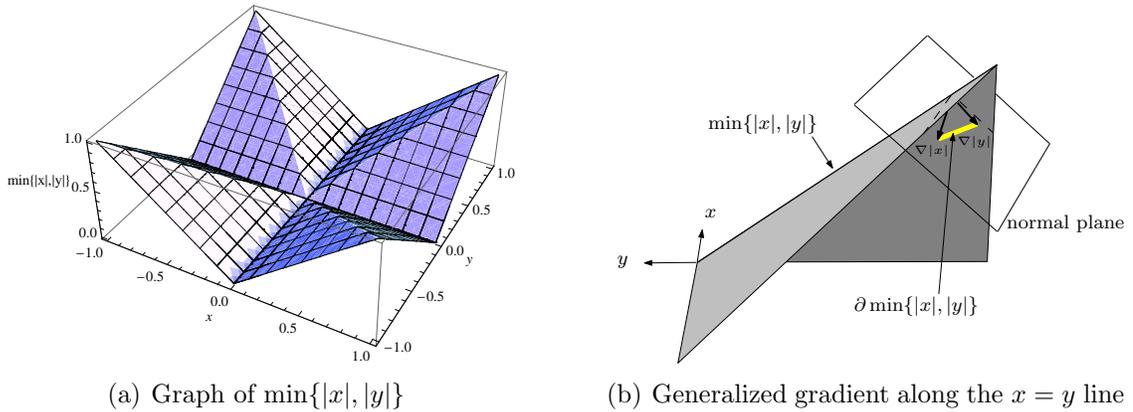


Figure 4.2: The $\min\{|x|, |y|\}$ function and its generalized derivative. Figure 4.2(a) shows that the function has minima along the x and y axes. Along the lines $x = y$ and $x = -y$ the function is not differentiable. Figure 4.2(b) illustrates why the generalized gradient along the $x = \pm y$ lines does not contain zero: $(0, 0) \notin \text{co}\{(0, 1), (1, 0)\}$! There is no positive λ for which $(0, 0) = \lambda(0, 1) + (1 - \lambda)(1, 0)$.

The generalized gradient of f_1 away from $x = 0$ is given by

$$x \neq 0 \Rightarrow \nabla|x| = \begin{cases} (1, 0), & x > 0 \\ (-1, 0), & x < 0 \end{cases}$$

while for f_2 we have

$$y \neq 0 \Rightarrow \nabla|y| = \begin{cases} (1, 0), & y > 0 \\ (-1, 0), & y < 0 \end{cases}.$$

Therefore, away from the axes and the lines $x = \pm y$, the generalized gradient

of f will be a singleton:

$$\nabla f(x, y) = \begin{cases} (0, 1), & 0 < x < y \\ (1, 0), & 0 < y < x \\ (-1, 0), & -y < x < 0 \\ (0, 1), & x < -y < 0 \\ (0, -1), & x < y < 0 \\ (-1, 0), & y < x < 0 \\ (1, 0), & 0 < x < -y \\ (0, -1), & 0 < -y < x. \end{cases}, \quad x \neq \pm y, x, y \neq 0 .$$

It can now be seen (and Fig. 4.2(b) illustrates), that along the lines $x = \pm y$ for $x, y \neq 0$, ∂f is a convex set that does not contain the origin.

The distance function $d(q)$ is practically a generalization of $f(x, y)$ in multiple dimensions and behaves very similarly in terms of its (generalized) derivatives. A notable property of ∂d (which can be verified in the example) is the following:

Lemma 2. *Let $d(q)$ be defined by (4.10). Then for $q \neq 0$, and for any z, w such that $z \in \partial d(q)$ and $w \in -\partial d(q)$, we have $\langle z, w \rangle \leq 0$. In addition, $\langle z, w \rangle = 0$ only when $z \in \delta \partial d$ and $w \in \delta \partial d$.*

Proof. The generalized gradient of $d(q)$ is the convex hull of unit vectors e_i along different coordinate directions. Any $z \in \partial d(q)$ is thus written as $z = \sum a_i e_i$, with $\sum a_i = 1$ and $a_i > 0$, while $w = \sum b_i (-e_i)$, with $\sum b_i = 1$ and $b_i > 0$, where i takes values in a finite set $\{1, \dots, k\}$ for some k . Therefore, $\langle z, w \rangle = \sum_{i,j \in \{1, \dots, k\}} (-a_i b_j) \langle e_i, e_j \rangle$, where we note that $\langle e_i, e_j \rangle = 1$ if $i = j$ and zero otherwise. Since $a_i, b_j \geq 0$, the sum is negative semidefinite. For $\langle z, w \rangle = 0$ we need to have $(-a_i b_i) \langle e_i, e_i \rangle = 0 \Rightarrow a_i b_i = 0$. For all i therefore, either a_i or b_i are

zero, implying that in this case both z and w are on the boundary of ∂d and $-\partial d$, respectively. \square

For functions expressed as pointwise minima (the case of maxima can be treated similarly) where Proposition 2 applies, we know the following:

Lemma 3 ([13]). *The origin is contained in the interior of the convex hull of a set of n arbitrary vectors $\{v_i \in \mathbb{R}^m, i = 1, \dots, n\}$ iff there exists a v_i such that for all $w \in \mathbb{R}^m$, $\langle w, v_i \rangle > 0$.*

Chapter 5

A NEW CONSTRUCTION FOR MULTI-AGENT NAVIGATION

In this chapter we introduce the new construction of the multi-agent navigation function and its properties. We have seen in Chapter 3 the limitations involved in extending the case of single agent navigation function to multiple agents using the existing approach. We therefore define a different structure for the navigation function for multiple agents for which we provide a proof of correctness.

5.1 Construction

We follow a similar construction for the goal function as in the case of a single agent navigation function, however the obstacle function differs, in that, we no longer use the product of the individual beta functions. We use a non-differentiable obstacle function arising from the need to limit the interaction between the agents to be local.

We treat the agents as autonomous identical sphere-shaped agents of diameter d_0 . For the sake of simplicity we do not consider any static obstacles to be present in the environment. This makes the collisions to occur only among the agents. At the cost of not being able to generalize the proposed method to star-shaped words, we relax the requirement for having analytic and admissible functions. It appears that there can be benefits in using nonsmooth functions to construct φ , which we intend to exploit.

The goal is to steer the agents from any relative initial configuration so that they fall into a pre-specified formation, without colliding with each other and without fixing the location of that formation in space. The formation itself is described by means of a graph:

5.2 The formation graph

The final formation that needs to be achieved by the agents is represented in the form of a directed graph called formation graph. The relative distances between the agents can be decoded from it. The absolute positions of the agents can then be known by the application of Moore-Penrose pseudo inverse on the relative distances vector.

Definition 16 (Formation graph [43]). *The formation graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{C}\}$ is a directed labeled graph consisted of:*

- *a set of vertices $\mathcal{V} = \{v_1, \dots, v_N\}$, indexed by the mobile agents,*
- *a set of edges $\mathcal{E} = \{(i, j) \in \{1, \dots, N\} \times \{1, \dots, N\}\}$, containing ordered pairs of nodes that represent inter-agent position specifications, and*
- *a set of labels (formation specifications) $\mathcal{C} = \{c_{ij} \mid (i, j) \in \mathcal{E}\}$.*

Whenever there is a $c_{ij} \in \mathcal{C}$, it implies that the desired relative position between agent i and agent j is c_{ij} , that is, ideally, we should have $\|p_i - p_j - c_{ij}\| = 0$. If graph \mathcal{G} is (weakly) connected, then the formation is uniquely specified.

Meeting the formation specifications depends on relative positions only, and our analysis is therefore performed in the space of relative differences.

$$q_{ij} \triangleq p_i - p_j .$$

If we denote p the stack vector of absolute agent positions, and q the stack vector of relative agent positions (differences), then

$$q = Bp,$$

where B is the incidence matrix of graph \mathcal{G} . Shifting the analysis to the space of relative positions (exclusively) avoids mixing absolute and relative coordinates (e.g., when considering both formation specifications and static obstacle avoidance), which leads to both analytic and conceptual problems when it comes to establishing the properties of a potential function.

Since the maximum number of possible edges in a graph of size N is $\frac{N(N-1)}{2}$, the workspace in this case is

$$\mathcal{Q} = \mathbb{R}^{N \times \frac{N(N-1)}{2}} - \{q \mid \|q\| > \rho_0, \|q_{ij}\| \leq 2d_0, \forall (i, j) \in N \times N\} ,$$

that is, it excludes collision configurations between agents and configurations where agents are too far apart from each other to be regarded as a group. We assume a decomposition of \mathcal{Q} similar to the decomposition of \mathcal{F} . We will work with a function φ , which is not a navigation function in the strict sense, but it is just polar, Morse, and it blows up at the boundary of the collision free space:

$$\varphi(q) = \frac{\gamma(q)}{\beta(q)} , \tag{5.1}$$

where $\gamma(q)$ is the goal function, and $\beta(q)$ is the obstacle function.

5.3 The goal function

We extend the goal function defined in [38] to multiple agents by using the sum of the norms of the differences between relative position vectors and their corresponding formation specifications. The goal function thus captures the proximity of the current group configuration to the desired formation configuration.

We define the goal function $\gamma : \mathbb{R} \rightarrow \mathbb{R}^+$ as

$$\gamma(q) = \gamma_d(q)^\kappa , \tag{5.2}$$

where $\gamma_d(q)$ is a scalar valued positive definite function which is a measure of how close the current group configuration is to the desired formation and κ is a tuning

parameter, used to change the slope of the goal function $\gamma(q)$. The value of κ can be changed to alter the location and nature of the critical points in the workspace.

Function $\gamma_d(q)$ is thus defined as

$$\gamma_d(q) \triangleq \sum_{(i,j) \in \mathcal{E}} \|q_{ij} - c_{ij}\|^2 . \quad (5.3)$$

5.4 The obstacle function

We define the obstacle function for multiple agents as

$$\beta = \log(\mu - a e^{-(-r+d+d^2)^2}) , \quad (5.4)$$

with $\mu > 2$, and a, r positive scalar parameters that can be chosen to determine the location where β vanishes, and set its derivative at this location. Specifically, if one needs β to vanish when $d = d_0$, *i.e.*, on the boundary of the obstacle, and have a derivative equal to ζ there, then the choice

$$r = d_0^2 + d_0 + \frac{\zeta}{2(1-\mu)(1+2d_0)} \quad (5.5a)$$

$$a = (\mu - 1) e^{([d_0^2 + d_0 - r]^2)} \quad (5.5b)$$

meets the requirement. The argument of β is the minimum distance between agents

$$d(q) \triangleq \min_{ij} \{\|q_{ij}\|\} , \quad (5.6)$$

measured between the centers of their spherical shapes. The minimum is taken over every combination of $i, j \in \{1, \dots, N\}$. If the agents' radii are equal to ρ , then it follows that one needs to choose r and a so that β vanishes when $d = 2\rho$. The properties of this distance function which were discussed in some detail in Section 4.5 are important in the present analysis.

5.5 Properties of the goal and obstacle functions

5.5.1 Goal function properties

The goal function is a scalar positive definite function which turns zero only when the agents are at their destination configuration. Decrease in the parameter κ , “sharpen” the graph of γ towards the destination. Figure 5.1 illustrates this.

5.5.2 Obstacle function properties

The β function is a positive definite scalar function of the distance between agents, varying in the interval $[0,1]$, that vanishes when any two agents are in contact with each other. The choice of β as in (5.4) gives the obstacle function the following attributes.

1. It vanishes whenever any two agents collide and remains positive otherwise;
2. It approaches a constant asymptotically as agents grow further apart;
3. It is non-differentiable.

The need to resort to a non-differentiable function comes from the requirement to establish convergence properties for φ while maintaining the local nature of interaction between agents. This requirement is refined into the following list of conditions for β . Within the range of differentiable functions, one can verify that these conditions cannot be all satisfied:

1. $q \rightarrow \partial\mathcal{F} \Rightarrow \beta \rightarrow 0$: at collision configurations, β needs to vanish.
2. As $\|q\|$ increases, $\beta \rightarrow \text{constant} < \infty$, because as agents grow apart their interaction is supposed to weaken.
3. $\beta > 0$ for $q \in \mathcal{F}$.
4. $\frac{\partial\beta}{\partial x} > 0$ for β to serve as a metric of distance to collision configurations.

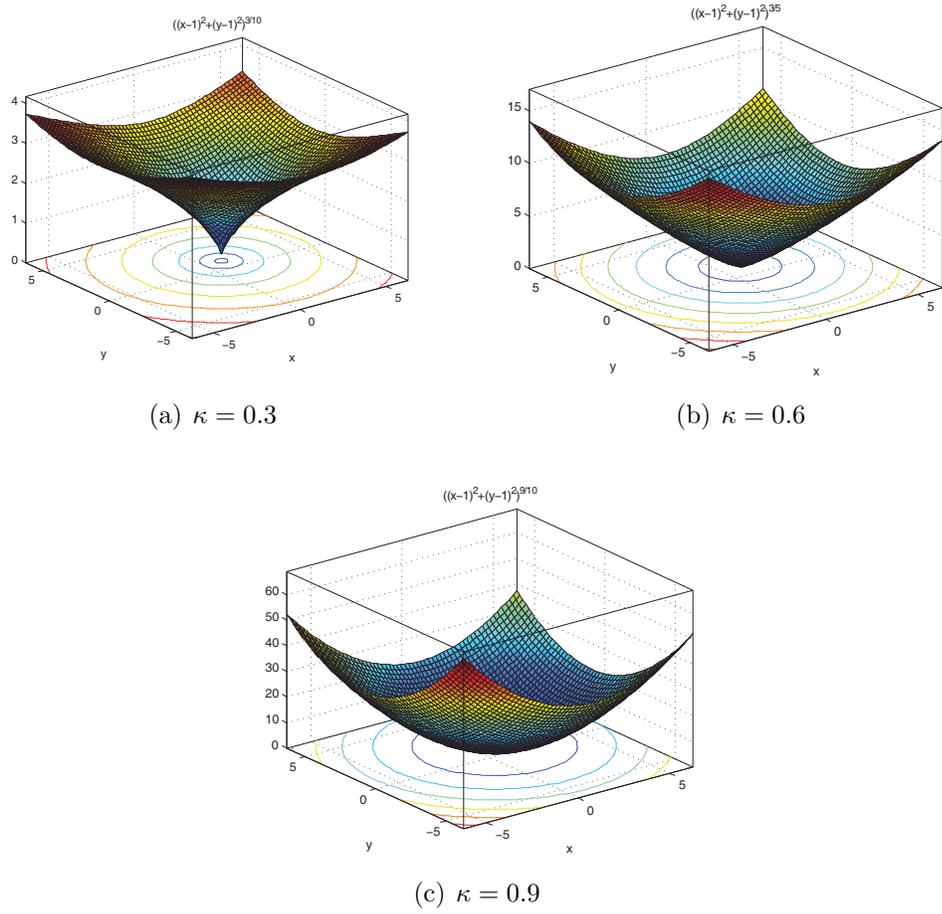


Figure 5.1: Variation in the shape of the goal function γ with the tuning parameter κ . The destination for the model goal function $((x - 1)^2 + (y - 1)^2)^\kappa$ is $(1, 1)$ Fig. 5.1(a) shows the graph for $\kappa = 0.3$, where the shape of the function is more sharpened towards the destination. Fig. 5.1(b) shows the same for $\kappa = 0.6$. For higher κ , the shape of γ looks smooth towards the destination as can be seen in Fig. 5.1(c)

5. $\nabla\beta \neq 0$ at $\partial\mathcal{F}$ so that we do not end up with critical points on, or have to slide along the boundary of the free space (needed in Proposition 6).
6. $\frac{\partial^2\beta}{\partial x^2} > 0$ near the critical points, to avoid running into the situation illustrated in (3.12).

One can observe that the requirement for β being strictly increasing and bounded in $[0, 1]$ is inconsistent with the specification that its second derivative is positive, at least for differentiable functions; the curvature has to switch. It may be the case that these particular conditions are not necessary in the form stated above, but it is not obvious how to relax them either. On the other hand, one might give up the requirement for a bounded β , forcing the agents to “feel” each other’s influence anywhere they are in their workspace (no localized collision avoidance interaction). In this work, we attempt to address the requirement for localized interaction directly, by adopting a nonsmooth structure for β that fits within the specifications set above.

In view of (5.1) and (4.8), the proposed potential function can be expressed as

$$\varphi(q) = \max_{\substack{i \in \{1, \dots, N\} \\ j \neq i}} \left\{ \frac{\gamma(q)}{\beta(\|q\|)} \right\} . \quad (5.7)$$

It becomes evident from the structure of the navigation function in (5.7), that it is nonsmooth. As it can be imagined, in the case where two or more agents achieve the same minimum β value, the max function would generate two or more values, the gradient for which cannot be computed using the differential tools for smooth functions. The new nonsmooth navigation function thus requires the results from Chapter 4 on nonsmooth analysis.

Chapter 6

PROOF OF CORRECTNESS

In this chapter the proof of correctness for the new construction of the navigation function is discussed in detail. The chapter is divided into two sections, the first of which deals with the case when the distance function is differentiable and the other when the distance function is non-differentiable. The results on nonsmooth analysis from Chapter 4 are used in the latter case.

6.1 When the distance function is differentiable

This section demonstrates the proofs for the situations where the distance functions is differentiable, namely away from configurations where inter-agent distances become equal to each other, and away from configurations where $q_{ij} = 0$. The latter configurations are infeasible since the agents are assumed to have a non-zero spherical volume, and a configuration where $q_{ij} = 0$ corresponds to a situation where they overlap in their workspace.

6.1.1 The destination is non-degenerate

The following proposition establishes the fact that the destination configuration q_d is a non-degenerate local minimum of φ . This is important because the destination configuration needs to be isolated.

Proposition 4. *(cf. [28, Proposition 3.1]) If the workspace is valid, the destination point, q_d , is a non-degenerate local minimum of φ .*

Proof. We have

$$\varphi = \frac{\gamma}{\beta} ,$$

where $\beta = \log(\mu - a e^{-(-r+d+d^2)^2})$. To make the destination configuration q_d non-degenerate we apply the following diffeomorphic transformation on the navigation function φ ,

$$\sigma(x) = (x)^{1/k} .$$

The result of the above transformation on φ would be

$$\bar{\varphi} = \frac{\gamma_d(q)}{\beta^{1/k}} .$$

We use the result from the Proposition 2.7 of [28], which states that the transformation of the navigation function φ with a monotonically increasing function such as $\sigma(x)$ produces a function $\bar{\varphi}$ whose critical points are same as that of φ . We state the proposition below.

Proposition 5 (Proposition 2.7 [28]). *Let $I_1, I_2 \subset \mathbb{R}$ be intervals, $\varphi : F \rightarrow I_1$ and $\sigma : I_1 \rightarrow I_2$ be analytic. Define the composition $\hat{\varphi} : F \rightarrow I_2$, to be*

$$\hat{\varphi} \triangleq \sigma \circ \varphi .$$

If σ is monotonically increasing on I_1 , then the set of critical points of $\hat{\varphi}$ and φ coincide, i.e

$$\mathcal{C}_\varphi = \mathcal{C}_{\hat{\varphi}}$$

and the index of each point is identical,

$$\text{index}(\varphi)|_{\mathcal{C}_\varphi} = \text{index}(\hat{\varphi})|_{\mathcal{C}_{\hat{\varphi}}}$$

We, therefore, use $\bar{\varphi}$ to prove that q_d is non-degenerate.

At configurations where d is differentiable, we can write

$$\begin{aligned} \nabla \bar{\varphi} &= \frac{1}{\beta^{2/k}} (\beta^{1/\kappa} \nabla \gamma_d - \gamma_d \nabla \beta^{1/\kappa}) \\ &= \frac{1}{\beta^{2/\kappa}} \left(\beta^{1/\kappa} \nabla \gamma_d - \gamma_d \frac{1}{\kappa} \nabla \beta^{1/\kappa-1} \right) . \end{aligned} \tag{6.1}$$

Evaluating $\nabla\bar{\varphi}$ at the destination q_d gives

$$\nabla\bar{\varphi}(q_d) = \frac{\nabla\gamma_d(q_d)}{\beta(q_d)^{1/k}} .$$

The Hessian of $\bar{\varphi}$, on the other hand, is

$$\begin{aligned} \nabla^2\bar{\varphi} &= \frac{1}{\beta^{4/\kappa}} \left[\beta^{2/\kappa} \nabla(\beta^{1/\kappa}\nabla\gamma_d - \gamma_d\nabla\beta^{1/\kappa}) \right. \\ &\quad \left. - (\beta^{1/\kappa}\nabla\gamma_d - \gamma_d\nabla\beta^{1/\kappa})\nabla(\beta^{2/\kappa}) \right] \\ &= \frac{1}{\beta^{4/\kappa}} \left[\beta^{2/\kappa} (\beta^{1/\kappa}\nabla^2\gamma_d + \nabla\beta^{1/\kappa}\nabla\gamma_d^T \right. \\ &\quad \left. - \nabla\gamma_d\nabla\beta^{1/\kappa T} - \gamma_d\nabla^2\beta^{1/k}) \right. \\ &\quad \left. - (\beta^{1/\kappa}\nabla\gamma_d - \gamma_d\nabla\beta^{1/\kappa})2\beta^{1/\kappa}\nabla\beta^{1/\kappa T} \right] \end{aligned}$$

At q_d , we have $\nabla\gamma_d(q_d) = 0$ and $\gamma_d(q_d) = 0$. We thus have

$$\nabla\bar{\varphi}(q_d) = 0 ,$$

and the Hessian therefore reduces to

$$\nabla^2\bar{\varphi} = \frac{1}{\beta^{4/\kappa}} [\beta^{1/\kappa}\nabla^2\gamma_d] = \frac{\nabla^2\gamma_d}{\beta^{\frac{3}{\kappa}-2}} .$$

Since $\nabla^2\gamma_d = 2I$ we have

$$\nabla^2\bar{\varphi} = \frac{2}{\beta^{\frac{3}{\kappa}-2}} I ,$$

where I here denotes the $\frac{N(N-1)}{2}$ -dimensional identity matrix. With $\beta(q_d) > 0$, it follows that q_d is a non-degenerate critical point. □

In the following proposition we show that critical points do not exist on the workspace boundary.

6.1.2 There are no critical points on the workspace boundary $\partial\mathcal{F}$

Proposition 6. (cf. [28, Proposition 3.2]) *If workspace is valid, all critical points of φ are in interior of free space \mathcal{F}_2 .*

Proof. Recalling (6.1), we have

$$\nabla\bar{\varphi} = \frac{\beta^{1/\kappa}\nabla\gamma_d - \gamma_d\nabla\beta^{1/\kappa}}{\beta^{2/\kappa}} = \frac{\kappa\beta\nabla\gamma_d - \gamma_d\nabla\beta}{\kappa\beta^{1+1/\kappa}} ,$$

note that on the boundary of the collision space (*i.e.*, on $\partial\mathcal{F}$) we have $\beta = 0$, while both γ_d and $\nabla\beta$ (the latter by choosing $\zeta > 0$) do not vanish. Therefore, as $q \rightarrow \partial\mathcal{F}$, the magnitude of $\nabla\bar{\varphi}$ blows up while it aligns with that of $\nabla\beta \neq 0$, establishing the transversality of $\nabla\bar{\varphi}$ on the boundary. \square

In the following proposition, we show that by varying the tuning parameter κ , the critical points can be pushed away from the free workspace \mathcal{F}_2 arbitrarily close to the collision configurations in \mathcal{F}_0 .

6.1.3 The critical points can be pushed toward collision configurations

Proposition 7. (cf. [28, Proposition 3.3]) *For every $\epsilon > 0$ there exists a positive integer $N(\epsilon)$ such that if $\kappa \geq N(\epsilon)$ then there are no critical points of φ in $\mathcal{F}_2(\epsilon)$.*

Proof. Let q be a critical point of φ . Since at a critical point $\nabla\varphi = 0$, from (6.1) it follows that at this configuration

$$\beta\nabla\gamma = \gamma\nabla\beta .$$

Substituting for γ from (3.2) we get

$$\beta\kappa\gamma_d^{\kappa-1}\nabla\gamma_d = \gamma_d^\kappa\nabla\beta \Rightarrow \beta\kappa\nabla\gamma_d = \gamma_d\nabla\beta .$$

Taking norms on both sides

$$\kappa\beta\|\nabla\gamma_d\| = \gamma_d\|\nabla\beta\| .$$

A sufficient condition for the above equality *not* to hold is

$$\kappa > \frac{\gamma_d \|\nabla\beta\|}{\beta \|\nabla\gamma_d\|} . \quad (6.2)$$

The gradient of the obstacle function defined in (5.4) is expressed as

$$\nabla\beta = \frac{2a(2d+1)(d+d^2-r)}{\mu e^{(d+d^2-r)^2} - a} \nabla d ,$$

where ∇d is the gradient of the distance function (4.8). We therefore have

$$\|\nabla\beta\| = \left| \frac{2a(2d+1)(d+d^2-r)}{\mu e^{(d+d^2-r)^2} - a} \right| \|\nabla d\| .$$

It can be verified that both $\|\nabla\gamma_d\|$ (as long as q_d is away from collision configurations) and $\|\nabla\beta\|$ and hence the right hand side of (6.2) are bounded in \mathcal{F}_2 . In fact, since $\nabla\beta$ is upper bounded everywhere in \mathcal{F}_2 and β attains a minimum of ϵ at the boundary of \mathcal{F}_2 , it follows that the ratio $\frac{\|\nabla\beta\|}{\beta}$ is upper bounded in \mathcal{F}_2 . In addition, since γ_d and $\|\nabla\gamma_d\|$ are continuous functions that must attain their extremum points in \mathcal{F}_2 , and given that as $q \rightarrow q_d$, $\frac{\gamma_d}{\|\nabla\gamma_d\|} \rightarrow 0$, the bound for κ on the right hand side of (6.2) is finite anywhere in \mathcal{F}_2 . \square

The next result is a critical one. It ensures that with an appropriate choice of parameters, the critical points that have been pushed toward the obstacles are not local minima; the Hessian has at least one negative eigenvalue there (later shown to be only one).

6.1.4 Critical points close to collision configurations are not minima

Proposition 8. (cf. [28, Proposition 3.4]) *For any valid workspace, there exists an $\epsilon_0 > 0$ such that φ has no local minimum in \mathcal{F}_0 as long as $\epsilon < \epsilon_0$.*

Proof. We first evaluate the Hessian of φ , as

$$\begin{aligned}
\nabla^2\varphi &= \frac{1}{\beta^4} [\beta^2 \nabla(\beta\nabla\gamma - \gamma\nabla\beta) - (\beta\nabla\gamma - \gamma\nabla\beta) \nabla(\beta^2)] \\
&= \frac{1}{\beta^4} [\beta^2 (\beta\nabla^2\gamma + \nabla\beta\nabla\gamma^T \\
&\quad - \nabla\gamma\nabla\beta^T - \gamma\nabla^2\beta) - (\beta\nabla\gamma - \gamma\nabla\beta)2\beta\nabla\beta^T] \\
&= \frac{1}{\beta^3} [\beta (\beta\nabla^2\gamma - \gamma\nabla^2\beta + \nabla\beta\nabla\gamma^T - \nabla\gamma\nabla\beta^T) \\
&\quad - 2(\beta\nabla\gamma - \gamma\nabla\beta)\nabla\beta^T] . \tag{6.3}
\end{aligned}$$

At the critical point since $\beta\nabla\gamma - \gamma\nabla\beta = 0$, (6.3) after plugging in (3.2) reduces to

$$\nabla^2\varphi = \frac{1}{\beta^2} [\beta\nabla^2\gamma_d^\kappa - \gamma_d^\kappa\nabla^2\beta] \tag{6.4}$$

$$= \frac{\gamma_d^{\kappa-2}}{\beta^2} [\kappa\beta (\gamma_d \nabla^2\gamma_d + (\kappa - 1)\nabla\gamma_d \nabla\gamma_d^T) - \gamma_d^2 \nabla^2\beta] . \tag{6.5}$$

Substituting (3.2) in the equation for vanishing gradient at the critical point we have,

$$\kappa\beta\nabla\gamma_d = \gamma_d\nabla\beta , \tag{6.6}$$

and taking the outer product on both sides,

$$(\kappa\beta)^2\nabla\gamma_d\nabla\gamma_d^T = \gamma_d^2\nabla\beta\nabla\beta^T \tag{6.7}$$

substituting for $\nabla\gamma_d\nabla\gamma_d^T$ from the above equation in (6.5) yields

$$\nabla^2\varphi = \frac{\gamma_d^{\kappa-1}}{\beta^2} \left[\kappa\beta\nabla^2\gamma_d + \left(1 - \frac{1}{\kappa}\right) \frac{\gamma_d}{\beta} \nabla\beta\nabla\beta^T - \gamma_d\nabla^2\beta \right] \tag{6.8}$$

From (6.6), by taking norms on both sides we have,

$$\kappa\beta = \frac{\gamma_d\|\nabla\beta\|}{\|\nabla\gamma_d\|}$$

substituting for $\kappa\beta$ in (6.8), we have

$$\begin{aligned}
\nabla^2\varphi &= \frac{\gamma_d^{\kappa-1}}{\beta^2} \left[\frac{\gamma_d\|\nabla\beta\|}{\|\nabla\gamma_d\|} \nabla^2\gamma_d + \left(1 - \frac{1}{\kappa}\right) \frac{\gamma_d}{\beta} \nabla\beta\nabla\beta^T - \gamma_d\nabla^2\beta \right] \\
&= \frac{\gamma_d^\kappa}{\beta^2} \left[\frac{\|\nabla\beta\|}{\|\nabla\gamma_d\|} \nabla^2\gamma_d + \left(1 - \frac{1}{\kappa}\right) \frac{1}{\beta} \nabla\beta\nabla\beta^T - \nabla^2\beta \right] . \tag{6.9}
\end{aligned}$$

Let \tilde{v} be any unit vector orthogonal to $\nabla\beta$, that is $\tilde{v}^T \cdot \nabla\beta = 0$ with $\|\tilde{v}\| = 1$. Then the quadratic form $\tilde{v}^T (\nabla^2\varphi) \tilde{v}$ expands to

$$\begin{aligned} \tilde{v}^T (\nabla^2\varphi) \tilde{v} &= \frac{\gamma_d^\kappa}{\beta^2} \tilde{v}^T \left(\frac{\|\nabla\beta\|}{\|\nabla\gamma_d\|} \cdot \nabla^2\gamma_d - \nabla^2\beta \right) \tilde{v} \\ &= \frac{\gamma_d^\kappa}{\beta^2} \tilde{v}^T \left(2I \frac{\|\nabla\beta\|}{\|\nabla\gamma_d\|} - \nabla^2\beta \right) \tilde{v} , \end{aligned} \quad (6.10)$$

where I is the $\frac{N(N-1)}{2}$ -dimensional identity matrix. Now, the right hand of (6.10) to be negative, the following condition suffices

$$\max \left\{ 2 \frac{\|\nabla\beta\|}{\|\nabla\gamma_d\|} \right\} - \min \{ \sigma(\nabla^2\beta) \} < 0 \quad (6.11)$$

where $\sigma(\cdot)$ denotes the spectrum of a matrix. Toward this end, we recall (5.4),

$$\beta = \log(\mu - a e^{-(r+d+d^2)^2})$$

and we treat it as a function of two (not necessarily independent) variables, $x_1 \triangleq d$ and $x_2 \triangleq d^2$: $\beta = \beta(x_1, x_2)$. Then the first partial derivative of β with respect to q can be written

$$\nabla\beta = \frac{\partial\beta}{\partial x_1} \frac{\partial x_1}{\partial q} + \frac{\partial\beta}{\partial x_2} \frac{\partial x_2}{\partial q} .$$

while the second is

$$\nabla^2\beta = \frac{\partial^2\beta}{\partial x_1^2} \left(\frac{\partial x_1}{\partial q} \left(\frac{\partial x_1}{\partial q} \right)^T \right) + \frac{\partial\beta}{\partial x_1} \left(\frac{\partial^2 x_1}{\partial q^2} \right) + \frac{\partial^2\beta}{\partial x_2^2} \left(\frac{\partial x_2}{\partial q} \left(\frac{\partial x_2}{\partial q} \right)^T \right) + \frac{\partial\beta}{\partial x_2} \left(\frac{\partial^2 x_2}{\partial q^2} \right)$$

With reference to (5.4) we have,

$$\frac{\partial\beta}{\partial x_1} = \frac{\partial\beta}{\partial x_2} = \frac{2a(x_1 + x_2 - r)}{\mu e^{(x_1+x_2-r)^2} - a} > 0 , \quad (6.12)$$

and

$$\frac{\partial^2\beta}{\partial x_1^2} = \frac{\partial^2\beta}{\partial x_2^2} = \frac{2a[\mu(4r(x_1 + x_2) - 2(x_1 + x_2)^2 - 2r^2 + 1)e^{(x_1+x_2-r)^2} - a]}{(a - \mu e^{(x_1+x_2-r)^2})^2} ,$$

which for $(x_1+x_2) \rightarrow r^+$, in the region where the critical point is expected, converges to $\frac{2a}{\mu-a} > 0$, for $a < \mu$.

To determine $\min \{\sigma(\nabla^2\beta)\}$, we first note that with the partial derivatives of β being positive, and write

$$\begin{aligned} \min \{\tilde{v}^T \nabla^2 \beta \tilde{v}\} &= \frac{\partial^2 \beta}{\partial x_1^2} \min \left\{ \tilde{v}^T \frac{\partial x_1}{\partial q} \left(\frac{\partial x_1}{\partial q} \right)^T \tilde{v} \right\} + \frac{\partial \beta}{\partial x_1} \min \left\{ \tilde{v}^T \frac{\partial^2 x_1}{\partial q^2} \tilde{v} \right\} \\ &\quad + \frac{\partial^2 \beta}{\partial x_2^2} \min \left\{ \tilde{v}^T \frac{\partial x_2}{\partial q} \left(\frac{\partial x_2}{\partial q} \right)^T \tilde{v} \right\} + \frac{\partial \beta}{\partial x_2} \min \left\{ \tilde{v}^T \frac{\partial^2 x_2}{\partial q^2} \tilde{v} \right\} . \end{aligned} \quad (6.13)$$

The first and third term in (6.13) involve rank-one matrices made of the same vector, and thus their minimum eigenvalue is zero. With this observation, (6.13) we write

$$\min \{\tilde{v}^T \nabla^2 \beta \tilde{v}\} \geq \frac{\partial \beta}{\partial x_1} \min \left\{ \tilde{v}^T \frac{\partial^2 x_1}{\partial q^2} \tilde{v} \right\} + \frac{\partial \beta}{\partial x_2} \min \left\{ \tilde{v}^T \frac{\partial^2 x_2}{\partial q^2} \tilde{v} \right\} ,$$

and given that $\frac{\partial \beta}{\partial x_1} = \frac{\partial \beta}{\partial x_2}$,

$$\min \{\tilde{v}^T \nabla^2 \beta \tilde{v}\} \geq \frac{\partial \beta}{\partial x_1} \left[\min \left\{ \tilde{v}^T \left(\frac{\partial^2 x_1}{\partial q^2} + \frac{\partial^2 x_2}{\partial q^2} \right) \tilde{v} \right\} \right]$$

Rewrite $x_1 = d = \min \|q_{ij}\| = \sqrt{\min q_{ij}^T q_{ij}}$ and name the relative vector q_{ij} with the minimum norm w for convenience. Since d is assumed to be differentiable, around the critical point it will hold:

$$\begin{aligned} \frac{\partial^2 x_1}{\partial q^2} &= \frac{\partial^2 \left(\sqrt{\|w\|^2} \right)}{\partial^2 q^2} = \frac{\partial}{\partial q} \left(\frac{\partial}{\partial q} \left(\sqrt{\|w\|^2} \right) \right) \\ &= \frac{\partial}{\partial q} \left(\frac{[0 \cdots 0 \ w \ 0 \cdots 0]^T}{\sqrt{\|w\|^2}} \right) = \frac{\partial}{\partial q} \left(\frac{[0 \cdots 0 \ w \ 0 \cdots 0]^T}{\|w\|} \right) \\ &= \frac{\|w\| \frac{\partial [0 \cdots 0 \ w \ 0 \cdots 0]^T}{\partial q} - [0 \cdots 0 \ w \ 0 \cdots 0]^T \frac{\partial \|w\|}{\partial q}}{\|w\|^2} \\ &= \frac{\|w\| \text{diag}\{0, \dots, 0, I_n, 0, \dots, 0\} - [0 \cdots 0 \ w \ 0 \cdots 0]^T \left(\frac{[0 \cdots 0 \ w \ 0 \cdots 0]^T}{\|w\|} \right)^T}{\|w\|^2} \\ &= \frac{1}{\|w\|} \text{diag} \left\{ 0, \dots, 0, \frac{\|w\|^2 I_n - ww^T}{\|w\|^2}, 0, \dots, 0 \right\} . \end{aligned}$$

On the other hand,

$$\frac{\partial^2 x_2}{\partial q^2} = \frac{\partial^2 \|w\|^2}{\partial q^2} = 2 \text{diag}\{0, \dots, 0, I_n, 0, \dots, 0\} .$$

Putting it together,

$$\begin{aligned} \min \tilde{v}^T \frac{\partial^2 \beta}{\partial q^2} \tilde{v} &\geq \frac{1}{\|w\|} \frac{\partial \beta}{\partial x_1} \min \left\{ \tilde{v}^T \text{diag} \left\{ 0, \dots, 0, 2\|w\|I_n + \frac{\|w\|^2 I_n - ww^T}{\|w\|^2}, 0, \dots, 0 \right\} \tilde{v} \right\} \\ &= \frac{1}{\|w\|} \frac{\partial \beta}{\partial x_1} \min \left\{ 2\|w\| + 1 - \tilde{v}^T \frac{ww^T}{\|w\|^2} \tilde{v} \right\} = 2 \frac{\partial \beta}{\partial x_1} \end{aligned}$$

With reference to (6.11), a sufficient condition for the quadratic form $\tilde{v}^T \nabla^2 \varphi \tilde{v}$ to be negative in \mathcal{F}_0 is that

$$\max_{\mathcal{F}_0} \frac{\|\nabla \beta\|}{\|\nabla \gamma_d\|} < \min_{\mathcal{F}_0} \left\{ \frac{\partial \beta}{\partial d} \right\}. \quad (6.14)$$

With $\frac{\partial d^2}{\partial q}$ growing as fast as $\|\nabla \gamma_d\|$, and with the magnitude of $\frac{\partial \beta}{\partial d}$ being regulated arbitrary through the choice of ζ and a , there is always an appropriate choice of parameters ζ and a so that (6.14) is satisfied. Practically, the further the desired formation encoded in γ_d is from \mathcal{F}_0 (the near collision configurations), the easier (6.14) is to satisfy. \square

To show that φ is Morse we use the same lemma as Lemma 1, which says that the non-singularity of a linear operator follows from the fact that its associated quadratic form is sign definite on complementary subspaces.

Let $\xi_q(v)$ denote $v^T \nabla^2 \varphi(q) v$, where $q \in \mathcal{F}_0(\epsilon)$, and v is a vector in the tangent space T_q of $\mathcal{F}_0(\epsilon)$ at q . Then we can show that the remaining nonnegative eigenvalues of the Hessian of φ are all positive. The following proposition establishes the Morse nature of φ .

6.1.5 The navigation function is Morse

Proposition 9. (cf. [28, Proposition 3.6]) *There exists an $\epsilon_2 > 0$ such that for every $\epsilon < \epsilon_2$ at each critical point of φ in $\mathcal{F}_0(\epsilon)$, there is a direct sum decomposition $T_q = \mathcal{P}_q \oplus \mathcal{N}_q$, for which $\xi_q|_{\mathcal{P}_q}$ is positive definite, $\xi_q|_{\mathcal{N}_q}$ is negative definite, and $\dim(\mathcal{P}_q) = 1$.*

Proof. Assume $q \in \mathcal{B}(\epsilon)$ where $\mathcal{B}(\epsilon) = \{q : 0 < \beta < \epsilon\}$. Define $\mathcal{P}_q = \text{span}\{\nabla\beta(q)\}$ and let \mathcal{N}_q be the orthogonal component of \mathcal{P}_q in T_q . In Proposition 8 it was shown $\xi_q|_{\mathcal{N}_q}$ is negative definite as long as $\epsilon < \epsilon_0$; the goal now is to show that $\xi_q|_{\mathcal{P}_q}$ is positive definite.

Let $v = \frac{\nabla\beta}{\|\nabla\beta\|}$. From (6.1) and (3.2), at a critical point we recall (6.6):

$$\kappa\beta\nabla\gamma_d = \gamma_d\nabla\beta \ .$$

Just as in the proof of Proposition 8, taking outer products in (6.6) yields (6.7), and substituting in (6.5) we get (6.8). Here, we instead take squared norms on both sides of (6.6) to solve for $\kappa\beta$ as follows

$$(\kappa\beta)^2\|\nabla\gamma_d\|^2 = \gamma_d^2\|\nabla\beta\|^2 \Rightarrow \kappa\beta = \frac{\gamma_d}{4\kappa\beta}\|\nabla\beta\|^2 \ , \quad (6.15)$$

exploiting the fact that $\|\nabla\gamma_d\|^2 = \gamma_d^2$. Substituting for $\kappa\beta$ as found in (6.15) in (6.5)

$$\begin{aligned} \nabla^2\varphi(q) &= \frac{\gamma_d^{\kappa-1}}{\beta^2} \left[\frac{\gamma_d}{4\kappa\beta}\|\nabla\beta\|^2\nabla^2\gamma_d + \left(1 - \frac{1}{\kappa}\right) \frac{\gamma_d}{\beta}\nabla\beta\nabla\beta^T - \gamma_d\nabla^2\beta \right] \\ &= \frac{\gamma_d^\kappa}{\beta^2} \left[\frac{\|\nabla\beta\|^2}{2\kappa\beta}I + \left(1 - \frac{1}{\kappa}\right) \frac{1}{\beta}\nabla\beta\nabla\beta^T - \nabla^2\beta \right] \end{aligned}$$

Evaluating $\xi_q|_{\mathcal{P}_q}$ with $v = \frac{\nabla\beta}{\|\nabla\beta\|}$

$$\begin{aligned} v^T\nabla^2\varphi v &= \frac{\gamma_d^\kappa}{\beta^2} \left[\frac{1}{2\kappa\beta}\|\nabla\beta\|^2 + \left(1 - \frac{1}{\kappa}\right) \frac{1}{\beta}v^T\nabla\beta\nabla\beta^Tv - v^T\nabla^2\beta v \right] \\ &= \frac{\gamma_d^\kappa}{\beta^2} \left[\frac{1}{2\kappa\beta}\|\nabla\beta\|^2 + \left(1 - \frac{1}{\kappa}\right) \frac{1}{\beta} \left(\frac{\nabla\beta^T\nabla\beta}{\|\nabla\beta\|} \right)^2 - v^T\nabla^2\beta v \right] \\ &= \frac{\gamma_d^\kappa}{\beta^2} \left[\frac{2\kappa-1}{2\kappa\beta}\|\nabla\beta\|^2 - v^T\nabla^2\beta v \right] \end{aligned}$$

Thus, $v^T\nabla^2\varphi v$ is positive, if

$$\min \left\{ \frac{2\kappa-1}{2\kappa\beta}\|\nabla\beta\|^2 \right\} - \max \{ v^T\nabla^2\beta v \} > 0 \quad (6.16)$$

Expanding $\nabla^2\beta$ as in the proof of Proposition 8,

$$\begin{aligned} \max \{v^T \nabla^2 \beta v\} &\leq \frac{\partial^2 \beta}{\partial x_1^2} \max \left\{ v^T \frac{\partial x_1}{\partial q} \left(\frac{\partial x_1}{\partial q} \right)^T v \right\} + \frac{\partial \beta}{\partial x_1} \max \left\{ v^T \frac{\partial^2 x_1}{\partial q^2} v \right\} \\ &\quad + \frac{\partial^2 \beta}{\partial x_2^2} \max \left\{ v^T \frac{\partial x_2}{\partial q} \left(\frac{\partial x_2}{\partial q} \right)^T v \right\} + \frac{\partial \beta}{\partial x_2} \max \left\{ v^T \frac{\partial^2 x_2}{\partial q^2} v \right\} \\ &\leq \frac{\partial^2 \beta}{\partial x_1^2} (1 + 4\|w\|^2) + \frac{\partial \beta}{\partial x_1} \left(\frac{2\|w\| + 1}{\|w\|} \right) , \end{aligned} \quad (6.17)$$

simplified by the fact that $\frac{\partial \beta}{\partial x_1} = \frac{\partial \beta}{\partial x_2}$ and $\frac{\partial^2 \beta}{\partial x_1^2} = \frac{\partial^2 \beta}{\partial x_2^2}$. The terms $\frac{\partial^2 \beta}{\partial x_1^2}$ and $\frac{\partial \beta}{\partial x_1}$ that appear in the right hand side of (6.17), are both positive and bounded when $q \in \mathcal{B}(\epsilon)$, with bounds dependent on parameters ζ and d_0 as indicated in (5.5). In view of (6.16), it therefore suffices to show that an appropriately small choice of ϵ can make the first term of (6.16) sufficiently large so as the whole difference is positive. Indeed, for a sufficiently small ϵ , $\frac{\partial^2 \beta}{\partial x_1^2}$ is decreasing in $\mathcal{B}(\epsilon)$ and can be upper bounded by its limit as $x_1 \rightarrow d_0$, whereas $\frac{\partial \beta}{\partial x_1}$ is increasing and can be upper bounded by its value at $d = \beta^{-1}(\epsilon)$. The corresponding bounds are

$$\begin{aligned} \max_{\mathcal{Q}_0} \frac{\partial \beta}{\partial x_1} &\leq 2(\mu e^{-\epsilon} - 1) \sqrt{\log \left(\frac{a}{\mu - e^\epsilon} \right)} \\ \max_{\mathcal{Q}_0} \frac{\partial^2 \beta}{\partial x_1^2} &\leq \frac{2a[\mu(1 - 2(d_0^2 + d_0 - r)^2)e^{(d_0^2 + d_0 - r)^2} + a]}{(a - \mu e^{(d_0^2 + d_0 - r)^2})^2}. \end{aligned}$$

Note that the former increases almost linearly with ζ (see (5.5)), while the latter decreases with ζ . To this end, note first that $\min \left\{ \frac{2\kappa - 1}{2\kappa} \frac{\|\nabla \beta\|^2}{\beta} \right\} > 0$ when $\kappa > \frac{1}{2}$, and that $\frac{2\kappa - 1}{2\kappa}$ is strictly increasing and upper bounded by 1 as $\kappa \rightarrow \infty$. The factor involving β and its gradient can be bounded as follows. First recall that

$$\nabla \beta = \frac{\partial \beta}{\partial x_1} \cdot \frac{\partial x_1}{\partial q} + \frac{\partial \beta}{\partial x_2} \cdot \frac{\partial x_2}{\partial q} = \frac{\partial \beta}{\partial x_1} \frac{w}{\|w\|} + 2 \frac{\partial \beta}{\partial x_2} w .$$

Then, given that $\frac{\partial\beta}{\partial x_1} = \frac{\partial\beta}{\partial x_2}$, write

$$\begin{aligned} \min \left\{ \frac{\|\nabla\beta\|^2}{\beta} \right\} &= \min_{\mathcal{F}_0} \left\{ \frac{1}{\beta} \left(\frac{\partial\beta}{\partial x_1} \right)^2 \left(2w + \frac{w}{\|w\|} \right)^2 \right\} \geq (4d_0^2 + 4d_0 + 1) \min \left\{ \frac{1}{\beta} \left(\frac{\partial\beta}{\partial x_1} \right)^2 \right\} \\ &= \frac{4d_0^2 + 4d_0 + 1}{\epsilon} \min_{\mathcal{F}_0} \left\{ \left(\frac{\partial\beta}{\partial x_1} \right)^2 \right\}, \end{aligned}$$

since $\max_{\mathcal{F}_0} \beta = \epsilon$. On the other hand, $\frac{\partial\beta}{\partial x_1}$ is lower bounded in \mathcal{F}_0 , with its value at $d = d_0$ being equal to $\frac{\zeta}{2d_0+1}$. Thus we have the first term of (6.16) rising quadratically with ζ , while the second increasing at most linearly in \mathcal{Q}_0 . Therefore, for sufficiently large ζ and small ϵ , (6.16) can be satisfied. \square

The above proposition 9 implies that if φ is tuned properly, that is, for a sufficiently large κ , then φ will be Morse, because at the critical points in the interior of the free space the Hessian of φ will have a single positive eigenvalue while all other will be negative. Together, Propositions 4 through 9 establish that φ is a Morse function which has a single minimum at the destination configuration. As a consequence, the solutions of the gradient field that φ produces (at least those solutions that can be defined in the classical form) lead almost any initial condition (with the exception of a zero-measure set) to the destination configuration.

The next section deals with the case where the classical gradient is not defined, that is, the function φ is nonsmooth due to the non-differentiability of the distance function. We show using the calculus of generalized gradients and differential inclusions introduced in Chapter 4 that there can still be no local minimum to trap the integral lines of the negated gradient of the navigation function, and that invariance theorems for nonsmooth systems can be applied to establish the (almost global) asymptotic stability of the destination.

6.2 When the distance function is not differentiable

To analyze the stability of the formation in the case when the distance function d is not differentiable, the results from Chapter 4 on nonsmooth analysis and invariance principle are utilized. Based on these invariance results, and the characterization of the generalized gradient of the distance function discussed in Section 4.6, we establish the (almost¹) global attractive properties of the desired formation configuration.

Without reconsidering the proofs given in the above section, which are applicable even here, we extend the proofs to the above propositions at the points of nondifferentiability.

When the right-hand side of the differential equation (3.10) is only piecewise continuous, the solution $p(t)$ (and therefore $q(t)$) for $t \geq 0$, cannot be defined in the classical sense anymore. Instead, it takes the form of a Filippov solution to the differential inclusion

$$\dot{q} = F(q(t)), q(t) \subset F \tag{6.18}$$

where $F(\cdot)$ is the Filippov set-valued map which in this case will be given by

$$F(q) = \overline{\text{co}} \left\{ \lim_{k \rightarrow \infty} Bu(q[k]) : q[k] \rightarrow q, q' \neq \mathcal{M} \right\}, \tag{6.19}$$

where $\overline{\text{co}}$ denotes the convex closure of the convex hull, $q[k]$ is any sequence of points at which φ is differentiable, B is the incidence matrix of the formation graph, and \mathcal{M} can be any set of measure zero. Such Filippov solutions are absolutely continuous curves $q : [0; \infty) \rightarrow \mathcal{F}$, which satisfy (6.18) for almost all $t \in [0, \infty)$. For the existence and uniqueness properties of Filippov solutions, we refer to [1], [18].

We can show that Propositions 4 through 9 hold for the nonsmooth case as well. In fact, the analysis is actually simplified in some cases using the calculus of

¹ With the exception of a set of configurations of measure zero, that include the unstable critical points of the potential function and their attraction regions.

generalized gradients. The following sections indicate how each one of the propositions of Section 6.1 extends to the nonsmooth case. The proofs of Section 6.1 are not repeated; instead we indicate why and how they can be adapted to include cases where generalized gradients have to be used.

6.2.1 Extension of Proposition 4

When the distance function is not differentiable at q_d , then according to Definition 26, since q_d is a non-smooth critical point, we have

$$\sum_i \lambda_i d\bar{\varphi}_i = \sum_i \lambda_i \frac{\nabla \gamma_d - \frac{\gamma_d}{\kappa \beta} d\beta_i}{\beta^{\frac{1}{\kappa}}} = 0 \quad , \quad (6.20)$$

where the subscript i has been dropped from β_i since all β_i involved in the sum are equal at the critical point. Because of (6.20), the second differential of the Lagrangian $L(q, \lambda)$ when evaluated at a critical point simplifies to

$$\sum_i \lambda_i d^2 \bar{\varphi} = \sum_i \lambda_i \frac{\beta^{\frac{1}{\kappa}} \nabla^2 \gamma_d - \gamma_d d^2 \beta_i^{\frac{1}{\kappa}}}{\beta^{\frac{2}{\kappa}}} \quad .$$

Since $\nabla^2 \gamma_d = 2I$, $\gamma_d(q_d) = 0$, and $\sum_i \lambda_i = 1$, the second differential of the Lagrangian reduces to $2\beta^{\frac{1}{\kappa}} I \neq 0$ and is therefore regular at q_d .

6.2.2 Extension of Proposition 6

Where $d(q)$ is not differentiable, at critical point q , and in view of (6.20), we can see that using the chain rule [16, Theorem 2.3.9(ii)] on $\partial\beta$, we get

$$\partial\varphi = \frac{1}{\kappa \beta^{1/\kappa}} \left(\kappa \nabla \gamma_d - \frac{\gamma_d}{\beta} \frac{\partial \beta}{\partial d} \partial d \right) \quad . \quad (6.21)$$

As one approaches collision configurations, $\beta \rightarrow 0$, (6.21) suggests that $\partial\varphi \rightarrow -\gamma_d (\kappa \beta^{\frac{1}{\kappa} + 1})^{-1} \frac{d\beta}{dd} \partial d$ which does not contain the zero vector. Therefore, the arguments of Proposition 6 apply.

6.2.3 Extension of Proposition 7

Now let $d(q)$ be non-differentiable at q . Then a necessary condition for q to be a critical point is $0 \in \partial\varphi$. Equation (6.21) hence yields the following necessary condition for the existence of a critical point:

$$\kappa \frac{2\beta}{\gamma_d \frac{\partial\beta}{\partial d}} (q - q_d) \in \partial d .$$

While the left hand side can be turned into an arbitrarily large number by an appropriate assignment to κ , ∂d is always lower and upper bounded. This is because away from the destination and collision configurations, β and $q - q_d$ are lower and upper bounded by ϵ and $\sqrt{\epsilon}$, respectively. In any domain containing q_d , on the other hand, the terms in the denominator, γ_d and $\frac{\partial\beta}{\partial d}$ upper bounded. Thus, repeating the argument, for a sufficiently large κ establishes the statement of Proposition 7, namely that for a sufficiently large κ the critical points of φ can be pushed in the ϵ -neighborhood of the collision configurations.

6.2.4 Extension of Proposition 8

To extend Proposition 8 to the case where d is not differentiable, we do not have to employ the Hessian of φ . We need to adapt and use a number of results from [13]; the next proposition is a dual to [13, Proposition 3], with slightly more relaxed conditions. Due to the relaxation, we provide a proof.

Proposition 10. [13, Proposition 3] *Let $V : \mathcal{Q} \rightarrow \mathbb{R}, q \rightarrow \max_i f_i(q)$ be a mapping where all f_i are smooth functions. If $0 \in \text{int}(\partial V(y))$ then y is a local minimum of V .*

Proof. If f_i are smooth, then they are regular and from the positive maxima theorem [16], and $\partial V(q) = \text{co} \{ \lim_{q_k \rightarrow q} \nabla f_i(q_k) : i \in I(q) \}$, where $I(q)$ is the set of indices for which $f_i(q) = V(q)$. Based on Lemma 3, for the origin to belong in the interior of $\partial V(q)$, it is necessary that there exists an i such that $\langle \nabla f_i(q), w \rangle > 0$, for every

$w \in T_q \mathcal{Q}$ (the tangent space of \mathcal{Q} at q). Following the same reasoning as in the proof of [13, Proposition 3], there must be a function f_i that increases along any direction w from point q . This implies that $V(q)$ is a local minimum. □

The proofs of the following two statements can be constructed in a straightforward manner, similarly to how the proposition above was established from [13, cf. Proposition 3].

Proposition 11 ([13], Proposition 4). *Let $V(q) = \max_i f_i(q)$ where all f_i are smooth functions. At a saddle point, V is nonsmooth, and the origin is contained in $\delta \partial V$.*

Proposition 12 ([13], cf. Proposition 5). *Let $V(q) = \max_i f_i(q)$ where all f_i are smooth functions. At a local maximum of V , $0 = \partial V$.*

In view of the above statements, there is a simple argument that shows why nonsmooth critical points of φ can only be saddles: note that the interior of a convex hull of $k \leq n$ vectors in an n -dimensional vector space is empty; all points are on the boundary. In the case of $d(q)$, even when every q_{ij} is one-dimensional, the number of any nontrivial different distances between agents cannot be more than the dimension of \mathcal{Q} , in other words, and with reference to (4.10), $|\{(i, j) : i, j \in \{1, \dots, N\} i \neq j\}| \leq \dim \mathcal{Q}$. Therefore at points where $d(q)$ is not differentiable, any point ∂d will be a boundary point. According to the chain rule for generalized gradients [16, Theorem 2.3.9(ii)], and based on (6.21),

$$\partial \varphi = \overline{\text{co}} \left\{ \frac{\kappa \beta \gamma_d^{\kappa-1} \nabla \gamma_d - \gamma_d^\kappa \frac{\partial \beta}{\partial d} \zeta}{\beta^2} : \zeta \in \partial d(q) \right\} \quad (6.22)$$

which means that $\partial \varphi$ is essentially ∂d , scaled by $-\frac{\gamma_d^\kappa}{\beta^2} \frac{\partial \beta}{\partial d}$ and translated by $\frac{\kappa \beta \gamma_d^{\kappa-1}}{\beta} \nabla \gamma_d$. Since all the points of $\partial \varphi$ are boundary points, according to Proposition 11, nonsmooth critical points of φ are necessarily saddles.

6.2.5 Extension of Proposition 9: The navigation function is Morse

To extend Proposition 9 to the case where d is not differentiable, we use the nonsmooth critical point condition as defined in Definition 26 and show that the conditions required for a nonsmooth critical point to be non-degenerate are met by φ . To prove the first condition (ND1), consider the linear combination of the set of differentials in $\left\{d\varphi_j(x_0) \mid j \in \hat{I}(x_0) \setminus \{i\}\right\}$, where x_0 is the nonsmooth critical point. We have the linear combination as,

$$\begin{aligned} \Sigma \lambda_i d\varphi_i &= \Sigma \lambda_i \frac{\beta d\gamma - \gamma d\beta}{\beta^2} \\ &= \Sigma \frac{d\gamma}{\beta} - \Sigma \lambda_i \frac{\gamma}{\beta^2} \frac{\partial \beta}{\partial d} D d d_i \\ &= \frac{\kappa \gamma_d^{\kappa-1}}{\beta} \Sigma \lambda_i - \gamma_d^\kappa \frac{\partial \beta}{\partial d} \Sigma \lambda_i d d_i \\ &= \frac{\gamma_d^\kappa}{\beta^2} \left[\kappa \beta \Sigma \frac{d\gamma_d}{\gamma_d} \lambda_i - \frac{\partial \beta}{\partial d} \Sigma \lambda_i d d_i \right] . \end{aligned}$$

In the above equation (6.23), the vectors in $\left\{-\frac{\partial \beta}{\partial d} \Sigma \lambda_i d d_i\right\}$ can be compared with the vectors x_1, x_2, \dots, x_n in the linearly independent set \mathcal{X} of Proposition 14, because the vectors in $d d_i$ are the unit vectors along the coordinate axes, and hence linearly independent.

Further, the coefficients of the vectors in the term $\kappa \beta \Sigma \lambda_i \frac{d\gamma_d}{\gamma_d}$ of equation (6.23) can be compared to the coefficients $\mu_1, \mu_2, \dots, \mu_n$ in Proposition 14. Since the coefficients $\kappa \beta \frac{d\gamma_d}{\gamma_d}$ are all positive, it follows from Proposition 14 that the differentials $d\varphi_i$ at a nonsmooth critical point of φ are always linearly independent.

According to Definition 26, at a nonsmooth critical point, we have

$$\sum_i \lambda_i d\varphi_i = \Sigma_i \lambda_i \frac{\beta d\gamma - \gamma d\beta_i}{\beta_i^2} = 0 . \quad (6.23)$$

The subscript to β has been neglected due to the fact that all the β_i 's are same at a non-smooth critical point. Simplifying and substituting for γ as γ_d^κ , we get

$$\kappa \beta d\gamma_d - \gamma_d \Sigma_i \lambda_i d\beta_i = 0 .$$

Taking inner product with γ_d on both sides, we have

$$\kappa\beta \|\mathrm{d}\gamma_d\|^2 - \gamma_d \langle \Sigma_i \lambda_i \mathrm{d}\beta_i, \mathrm{d}\gamma_d \rangle = 0 .$$

Using, $\|\mathrm{d}\gamma_d\|^2 = 4\gamma_d$, we get $\kappa\beta$ as

$$0 < \kappa\beta = \frac{1}{4} \langle \Sigma_i \lambda_i \mathrm{d}\beta_i, \mathrm{d}\gamma_d \rangle . \quad (6.24)$$

Now, consider a vector v in the tangent space $\hat{T}(x_0)$. According to Definition 27

$$\hat{T}(x_0) = \cap_{i \in \hat{I}(x_0)} \ker \mathrm{d}\varphi_i(x_0) ,$$

i.e.,

$$v \in \cap_i \ker \mathrm{d}\varphi_i = \cap_i \ker(\beta \mathrm{d}\gamma - \gamma \mathrm{d}\beta_i) .$$

Since v is orthogonal to $(\beta \mathrm{d}\gamma - \gamma \mathrm{d}\beta_i)$, we have

$$\langle v, \beta \kappa \gamma_d^{\kappa-1} \mathrm{d}\gamma_d - \gamma_d^{\kappa} \mathrm{d}\beta_i \rangle = 0 .$$

Substituting for γ as γ_d^{κ} , and simplifying we get

$$\langle v, \mathrm{d}\gamma_d \rangle = \frac{\gamma_d}{\kappa\beta} \langle \mathrm{d}\beta_i, v \rangle . \quad (6.25)$$

Noting that v is orthogonal to $\mathrm{d}\varphi_i$, the convex combination of the Hessians φ_i can be evaluated to be found as

$$\begin{aligned} v^T \Sigma \lambda_i \mathrm{d}^2 \varphi_i v &= v^T \Sigma \frac{\beta \mathrm{d}^2 \gamma - \gamma \mathrm{d}^2 \beta_i}{\beta_i^2} , \\ &= v^T \Sigma_i \lambda_i \frac{\mathrm{d}^2 \gamma}{\beta} v - v^T \Sigma_i \lambda_i \frac{\gamma_d^2 \beta_i}{\beta^2} v. \end{aligned} \quad (6.26)$$

Since, β_i is the same for all $i \in I(q_0)$, it can be factored out. Substituting for γ as γ_d^{κ} , we get

$$v^T \Sigma \lambda_i \mathrm{d}^2 \varphi v = v^T \frac{\mathrm{d}(\kappa \gamma_d^{\kappa-1} \mathrm{d}\gamma_d)}{\beta} v \Sigma_i \lambda_i - \frac{\gamma}{\beta^2} v^T \Sigma_i \lambda_i \mathrm{d}^2 \beta_i v$$

$$\begin{aligned}
&= \frac{\kappa}{\beta} v^T d(\gamma_d^{\kappa-1} d\gamma_d) v - \frac{\gamma}{\beta^2} v^T \Sigma_i \lambda_i d^2 \beta_i v \\
&= \frac{\kappa}{\beta} v^T \{ (\kappa - 1) \gamma_d^{\kappa-2} d\gamma_d d\gamma_d^T + \gamma_d^{\kappa-1} d^2 \gamma_d \} v \\
&\quad - \frac{\gamma}{\beta^2} v^T \Sigma_i \lambda_i d^2 \beta_i v \\
&= \frac{\kappa}{\beta} \gamma_d^{\kappa-2} v^T \{ (\kappa - 1) [d\gamma_d d\gamma_d^T] + \gamma_d 2I \} v \\
&\quad - \frac{\gamma_d^\kappa}{\beta^2} v^T \Sigma_i \lambda_i d^2 \beta_i v \\
&= \frac{\gamma_d^{\kappa-2}}{\beta^2} v \{ \kappa \beta ((\kappa - 1) [d\gamma_d d\gamma_d^T] + 2\gamma_d I) - \gamma_d^2 \Sigma_i \lambda_i d^2 \beta_i \} v \\
&= \frac{\gamma_d^{\kappa-2}}{\beta^2} \left\{ 2\kappa \beta \gamma_d I + \frac{(\kappa - 1) \gamma_d^2}{\kappa \beta} \|v^T d\beta_i\|^2 - \gamma_d^2 v^T \Sigma_i \lambda_i d^2 \beta_i v \right\} . \tag{6.27}
\end{aligned}$$

Substituting for $\kappa\beta$ from (6.24) and using (6.25) in the above equation, we get

$$\begin{aligned}
v^T \Sigma_i \lambda_i d^2 \varphi v &= \frac{\gamma_d^\kappa}{\beta^2} \left\{ 2 \frac{d\gamma_d^T \sum_i \lambda_i d\beta_i}{\gamma_d} I + \frac{(\kappa - 1) \|v^T d\beta_i\|^2}{d\gamma_d^T \sum_i \lambda_i d\beta_i} \right. \\
&\quad \left. - v^T \Sigma_i \lambda_i d^2 \beta_i v \right\} \tag{6.28}
\end{aligned}$$

From the above equation two cases arise

Case 1: When v is $\not\perp d\beta_i$

When $v \not\perp d\beta_i$, the value of κ can be raised to make sure that the sum of the first two terms in the brackets in equation (6.28) is large enough to make the whole term positive. This adjustment on κ is consistent with the proof of Proposition 9 for the smooth case.

Case 2: When v is $\perp d\beta_i$

When $v \perp d\beta_i$, then the middle term in the parenthesis in equation (6.28)

vanishes and we are left with the following equation.

$$v^T \sum_i d^2 \varphi v = \frac{\gamma_d^\kappa}{\beta^2} \left\{ 2 \frac{d\gamma_d^T \sum_i \lambda_i d\beta_i}{\gamma_d} I - v^T \sum \lambda_i d^2 \beta_i v \right\} \quad (6.29)$$

$$\frac{\gamma_d^\kappa}{\beta^2} \left\{ 2 \left(\frac{d\gamma_d}{\|d\gamma_d\|} \right)^T \frac{\sum \lambda_i d\beta_i}{\|d\gamma_d\|} - v^T \sum \lambda_i d^2 \beta_i v \right\} \quad (6.30)$$

$$\geq \frac{\gamma_d^\kappa}{\beta^2} \left\{ 2 \max_{\mathcal{F}_0} \frac{\|\sum \lambda_i d\beta_i\|}{\|d\gamma_d\|} - \min_{\mathcal{F}_0} \lambda(\sum \lambda_i d^2 \beta_i) \right\} , \quad (6.31)$$

where $\lambda(\cdot)$ denotes an eigenvalue. Obviously, if the right hand side of (6.31) is negative, the left hand side has to be also. Relating $\sum \lambda_i d\beta_i$ to $\nabla\beta$ and $\sum \lambda_i d^2 \beta_i$ to $\nabla^2\beta$, the similarity between (6.31) and (6.10) becomes apparent, and thus choosing ζ appropriately large along the lines of the proof of Proposition 8, makes $v^T \sum_{i \in \bar{I}} \lambda_i d^2 \varphi_i v$ negative definite.

Thus, (6.28) can be made non-singular—either positive or negative definite—by a proper selection of ζ . This makes the nonsmooth critical point a nondegenerate critical point.

It has been therefore shown that irrespectively of whether φ is differentiable or not, when appropriately tuned the construction given by (5.1)-(5.2)-(5.3)-(5.4) yields a Morse function with a single minimum at q_d ; with all other critical points being saddles. Remarkably, regions where φ is not differentiable can only produce saddles, but tuning is still necessary to adjust critical points at locations where φ is differentiable. We next show that properties established by Propositions 4 through 9 and their nonsmooth extensions are sufficient to ensure almost global asymptotic convergence of (6.18) to q_d .

6.3 Proof of convergence of φ

Let us first design the right hand side of (6.18) by setting

$$Bu = -\text{Ln}(\partial\varphi)(q) \quad (6.32)$$

where $\text{Ln}(S)(q)$ is a set valued map that assigns to each subset of S the set of least-norm elements in the closure of S [18]; in cases where S is a convex and closed (as for $S = \partial\varphi$), $\text{Ln}(S)(q)$ maps to a singleton which is the orthogonal projection of the zero vector on S . The right hand side of (6.18) takes the form of (6.32) if the agent inputs are chosen as

$$u = -B^\dagger \text{Ln}(\partial\varphi)(q) \quad , \quad (6.33)$$

where $B^\dagger = (B^T B)^{-1} B^T$ is the Moore-Penrose generalized inverse of the incidence matrix of the formation graph \mathcal{G} , B . With this choice of control inputs,

$$F(q) = -\partial\varphi(q) \quad , \quad (6.34)$$

that is, (6.18) takes the form of nonsmooth generalized gradient flow [18].

The set $\{V^o(z; v) | v \in F(z)\}$ in Theorem 1 can take values in a superset of the generalized Lie derivative $\bar{\mathcal{L}}_F(\partial f)$ of [18], [5], making the task of establishing the sign (semi)definiteness much harder when applying Theorem 1; arguably, there is a price for relaxing the regularity condition on V . In our case, however, φ is locally Lipschitz and regular because it is a pointwise maximum function [16, Proposition 2.3.12]. We can therefore bring to bear a much stronger result that directly applies to nonsmooth generalized gradient flows [18, Proposition 11]:

Proposition 13 ([18]). *Let $V : \mathbb{R}^N \rightarrow \mathbb{R}$ be locally Lipschitz and regular. Then the strict minimizers of V are strongly stable equilibria of the nonsmooth gradient flow of V . Furthermore, if the level sets of V are bounded, then the solutions of the nonsmooth gradient flow asymptotically converge to the set of critical points of V .*

In our case taking $\varphi(q)$ as V , we know that the level sets of φ are bounded in \mathcal{Q} . The convergence of the closed loop system (6.18)-(6.33) therefore follows from a direct application of Proposition 13.

An alternative proof using Theorem 1 is provided below.

6.4 Proof of convergence using Ryan's invariance principle

We take φ as $V(q)$. Function φ is continuous and locally Lipschitz. At every q where $d(q)$ is differentiable, $\dot{q} = -\nabla\varphi$, so $V^o(q, \dot{q}) = \dot{V}(q) = -\|\nabla\varphi\|^2$. At configurations q where $d(q)$ is non-differentiable, \dot{q} takes the form of Filippov solution to a differential inclusion, *i.e.*,

$$\dot{q} \in F(q) ,$$

where $F(q) = -\partial\varphi(q)$.

In such cases, we have

$$V^o(q, \dot{q}) \in \{w | \exists v \in \partial\varphi, \exists \zeta \in \partial\varphi(q) : \langle \zeta, v \rangle = w\} .$$

Given (6.22),

$$\begin{aligned} V^o(q, \dot{q}) &\in \left\{ w | \exists v, \zeta \in \partial d : -\frac{1}{\beta^4} (\kappa\beta\gamma_d^{k-1}\nabla\gamma_d - \gamma_d^\kappa \frac{\partial\beta}{\partial d} v)^T \cdot (\kappa\beta\gamma_d^{k-1}\nabla\gamma_d \right. \\ &\quad \left. - \gamma_d^\kappa \frac{\partial\beta}{\partial d} \zeta) = w \right\} \\ &= \left\{ w | \exists v, \zeta \in \partial d : -\frac{\gamma_d^{2\kappa-2}}{\beta^4} (\kappa^2\beta^2\|\gamma_d\|^2 + \gamma_d^2 (\frac{\partial\beta}{\partial d} v)^T (\frac{\partial\beta}{\partial d} \zeta)) \dots \right. \\ &\quad \left. - \gamma_d (\frac{\partial\beta}{\partial d} v)^T \kappa\beta\nabla\gamma_d \kappa\beta\gamma_d \nabla\gamma_d^T (\frac{\partial\beta}{\partial d} \zeta) \right\} \tag{6.35} \\ &= \left\{ w | \exists v, \zeta \in \partial d : -\frac{\gamma_d^{2\kappa-2}}{\beta^4} (4\kappa^2\beta^2\gamma_d - \gamma_d\beta \frac{\partial\beta}{\partial d} [v^T \nabla\gamma_d \nabla\gamma_d^T \zeta] \kappa \dots \right. \\ &\quad \left. + \gamma_d^2 (\frac{\partial\beta}{\partial d})^2 v^T \zeta) \right\} \end{aligned}$$

Rephrasing, there exist $\zeta \in \partial d$ and $v \in -\partial d$ for which

$$V^o(q, \dot{q}) = -\frac{\gamma_d^{2\kappa-1}}{\beta^4} (4\beta^2 \cdot \kappa^2 - \beta \frac{\partial\beta}{\partial d} [v^T \nabla\gamma_d + \nabla\gamma_d^T \zeta] \cdot \kappa + \gamma_d (\frac{\partial\beta}{\partial d})^2 v^T \zeta) ,$$

where we identify a second order polynomial in κ , having negative definite coefficient for κ^2 , and negative semidefinite (2) constant coefficient, everywhere in the interior

of \mathcal{Q} . The term multiplying κ , can be bounded in \mathcal{Q} using a tight bound that exploits the fact that ∂d is a closed orthant:

$$\beta \frac{\partial \beta}{\partial d} \left[v^T \nabla \gamma_d + \nabla \gamma_d^T \zeta \right] \leq \max_{\mathcal{Q}} \left\{ \beta \frac{\partial \beta}{\partial d} \right\} d^o(q; \nabla \gamma_d) \quad ,$$

and therefore there is an upperbound on κ for which $V^o(q; \dot{q})$ becomes negative semi-definite everywhere in $\text{int}(\mathcal{Q})$. Away from q_d , the locations where V^o vanishes are necessarily saddles due to Proposition 11, and therefore isolated.

The fact that $q(t)$ is a pre-compact solution of (6.18) is established by invoking the invariance of the level sets of φ : since the directional derivative of φ along the system's trajectories is non-positive, level sets that contain the initial configuration $q(0)$ are positively invariant (strongly, in this case, since no solution increases φ). With φ being positive definite in \mathcal{Q} once the coordinate system is shifted at q_d , that implies that the invariant level sets are also compact. The closedness of the solution $q(t)$ is given by definition (see [39]); the boundedness follows by the positive invariance of the level sets. According to Theorem 1 therefore, $q(t)$ approaches the largest weakly-invariant set in the level set of φ for which $y(q) = 0$. The points in \mathcal{Q} that satisfy $y(q) = 0$ are the saddle points of φ and q_d .

Chapter 7

NUMERICAL AND EXPERIMENTAL VALIDATION

In this chapter, we validate the proof of correctness of the navigation function with numerical and experimental results. We investigate the nature of the critical points of the proposed construction for φ , as the tuning parameter κ increases.

7.1 Numerical Results

We perform a side-by-side comparison of the new construction of the navigation function with a conventional construction as in (3.9). A three agent formation is the largest formation for which we can reasonably illustrate the location of critical points in two-dimensions. For a three agent formation, we identify the critical points of a potential function numerically. The computation of the critical points is done using standard numerical optimization methods available in MATLAB. This parallel numerical investigation provides evidence that both constructions may not give Morse functions for low values of their tuning parameter. Although the results appear contaminated by numerical noise at high values of the tuning parameter, giving several probable spurious critical points, they still seem to indicate that for low values of κ , there is *continuum* of critical points which excludes the possibility of them being non-degenerate.

7.1.1 Assessment of the critical points

Here we shall compare the traditional navigation function with the new construction on the basis of the behavior of critical points with change in the parameter κ . The standard routine `fmincon` of MATLAB is used.

7.1.1.1 Nature of the critical points - A comparison between the traditional construction and the new construction

In the traditional as well as the new construction, the desired formation configuration is set to $c_{21} = (0.2, 0.2)$ m, and $c_{31} = (0.3, 0.3)$ m. Figure 7.1 shows the critical points for different values of the tuning parameter, marking the location of the two agents in a coordinate system fixed on the third agent. In such a setup, the relative position of each of the other two agents can be mapped on the plane. Figure 7.2 depicts the location of critical points in the same setup for the proposed potential function construction, as the tuning parameter is increased.

In the case of Figure 7.1 we see a crescent-shape manifold of critical points appearing for small values of κ . In Figure 7.2, this manifold takes a circular shape surrounding the agent at which the local coordinate system is attached. As the parameter increases, however, the critical points seem to converge to q_d in both cases. The spurious critical points around q_d , and the few far away from it, are attributed to numerical noise; when the parameter κ increases, since the latter resides in the exponent of the denominator of φ , the function may exhibit dramatic variations.

7.1.2 Simulation of agent paths

This section presents numerical results for the agent paths with the agents starting from different initial positions. The relative positions between different agents is plotted against time. The simulation test assesses the behavior of the closed loop system under (6.33), in the neighborhood of a suspected saddle. In this

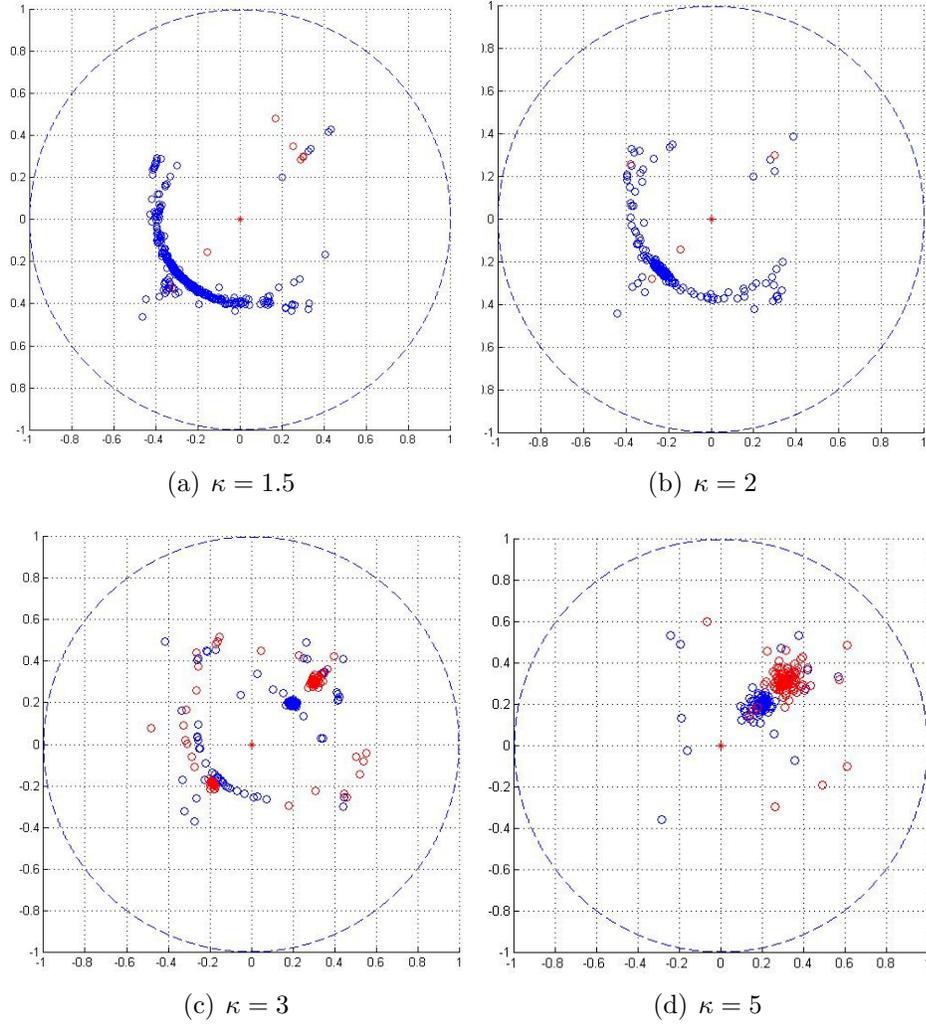


Figure 7.1: Numerical computation of critical points in a traditional multi-agent navigation function as given by (3.9). Agent 1 is assumed at the center, and the relative position of the other two agents when φ has a saddle is marked by a pair of small circles. Points $(0.2, 0.2)$ and $(0.3, 0.3)$ mark the desired configuration. For small values of κ a continuum of critical points appears; it later disappears as the parameter increases beyond a certain threshold.

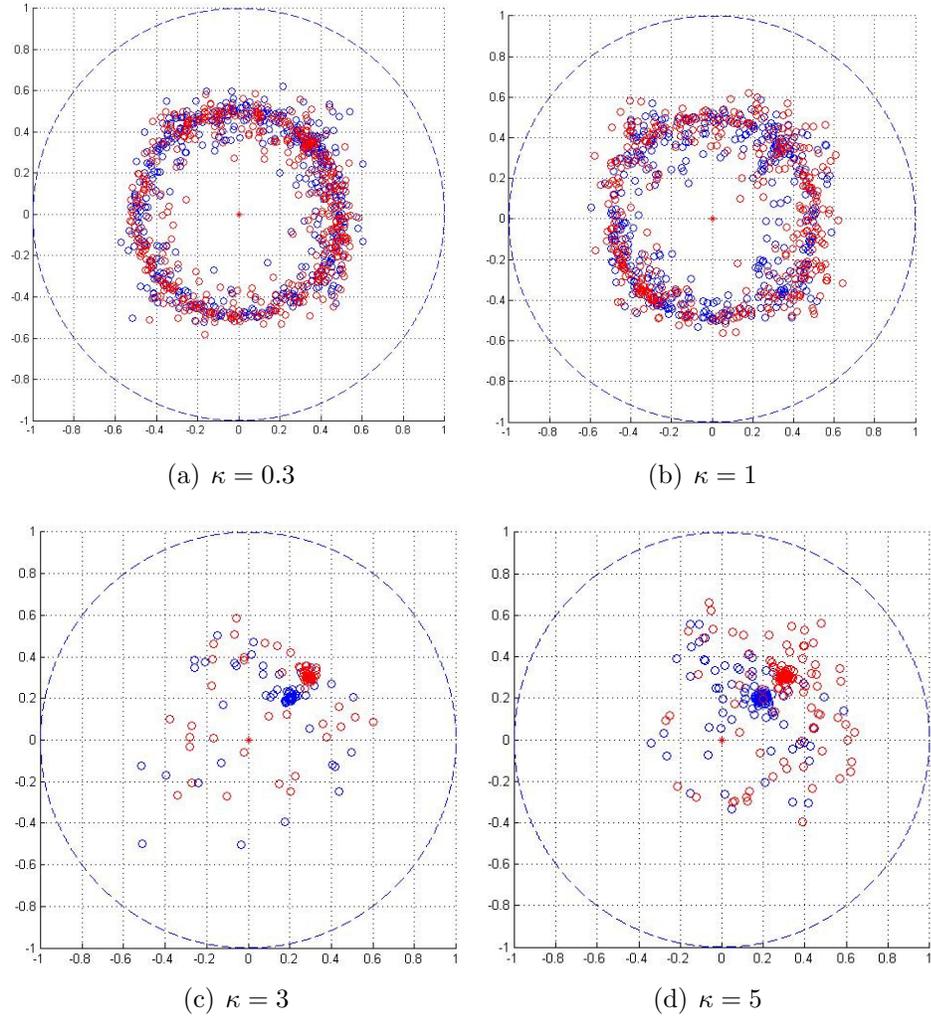


Figure 7.2: Numerical computation of critical points in the proposed multi-agent navigation function given by (5.1)-(5.2)-(5.3)-(5.4). Agent 1 is assumed at the center, and the relative position of the other two agents when φ has a saddle is marked by a pair of small circles. Points $(0.2, 0.2)$ and $(0.3, 0.3)$ mark the desired configuration. For small values of κ a ring of critical points appears and then disappears as the parameter increases beyond a certain threshold.

scenario we set $c_{12} = (0.1, 0.1)$ m and $c_{23} = (0.2, 0.2)$ m, so that the agents are expected to align along a 45 degree line, with agent 2 in the middle, agent 1 at the top right, and agent 3 at the bottom left.

In the first run we choose initial positions for the agents so that they are all along the 45 degree line, but at the “wrong” place: agent 1 is at the bottom left, agent 2 is at the top right, and agent 3 is in the middle. The agents have to swap relative positions to reach the desired configuration given by c_{12} and c_{23} defined earlier. The initial relative positions are hence given by $q_{12}(0) = (-0.3, -0.3)$ m, and $q_{23}(0) = (0.1, 0.1)$ m. As it may be expected, the symmetry of the initial configuration would force the agents to a deadlock; indeed, this is exactly what happens as shown in Fig. 7.3. Additional simulations have shown that the final relative position reached by the agents in Fig. 7.3 is also attractive for other initial conditions along the 45 degree line, with the agents in the same order but with different initial spacing. As shown next, the configuration where $x_{12} = y_{12} = -0.4$ and $x_{23} = y_{23} = 0.2$ is a saddle.

Figure 7.4 shows what happens when the formation is initialized slightly off the region of attraction of the saddle. Agents are able to swap positions, minimize their goal function asymptotically, and achieve their desired configuration.

Additional numerical simulations with initial conditions in the neighborhood of the saddle at $q_{12} = (-0.4, -0.4)$ m, $q_{23} = (0.2, 0.2)$ m exhibited the same behavior, confirming that the attraction region of that saddle is restricted to the straight line defined by the formation specification vectors c_{ij} given initially (Fig. 7.5). What Fig. 7.5 also confirms is that the fact that the agents swap positions with each other in the case of Fig. 7.4 (where agent 1 ends up at the initial position of agent 2, agent 2 goes where agent 3 initially was, and agent 3 converges to the initial location of agent 1) is just coincidental and due to the symmetry in the initial conditions; when the agents are initialized in the neighborhood of the saddle point, their final

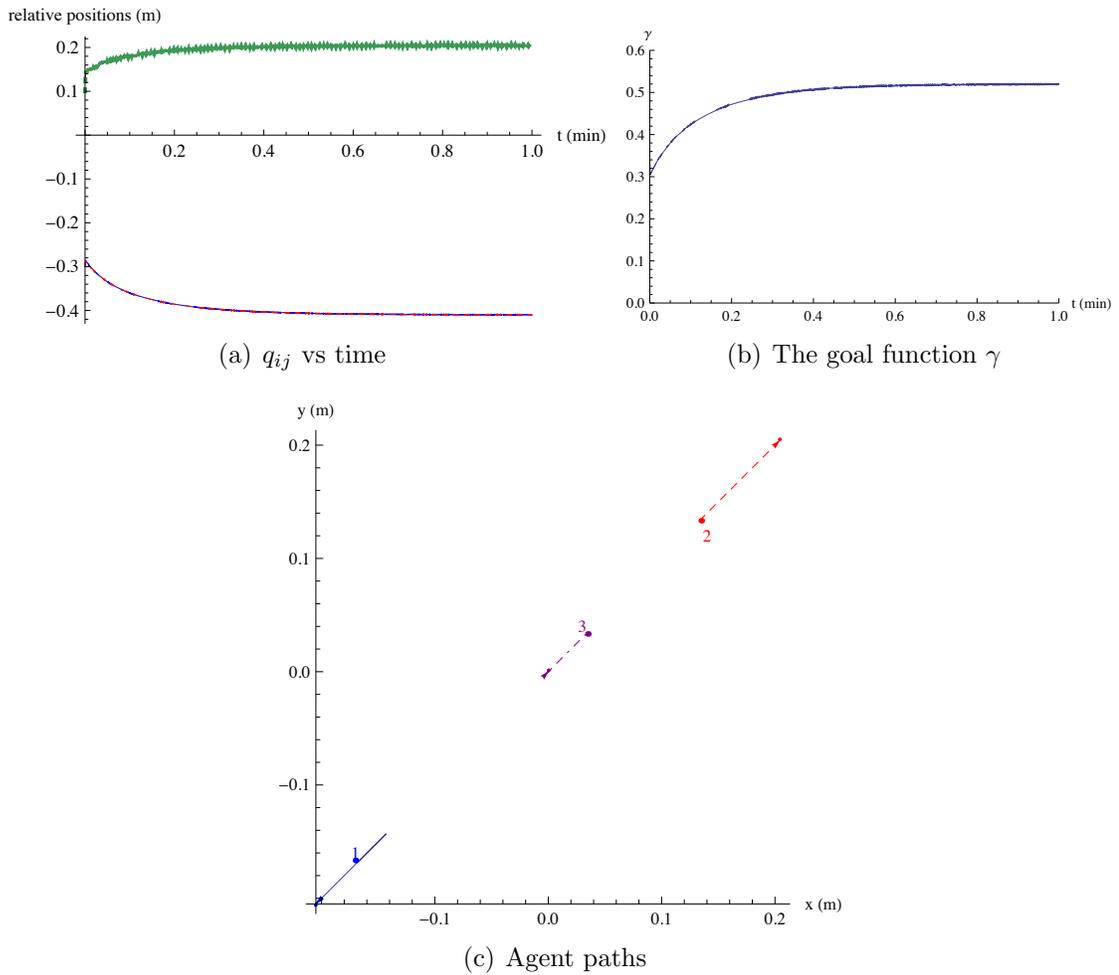


Figure 7.3: Simulation of a formation control maneuver where all agents have to switch positions, starting within the region of attraction of a saddle point which eventually traps them. In Figure 7.3(a) the continuous curve shows the evolution of $x_1 - x_2$, the dashed curve shows that of $y_1 - y_2$, the dot-dashed curve depicts $x_2 - x_3$, and the thick dotted one gives $y_2 - y_3$. Figure 7.3(b) reveals that the agents are not capable of minimizing their goal function γ . In Figure 7.3(c) the continuous path corresponds to that of agent 1, the dashed curve to agent 2, and the dot-dashed to agent 3. The initial positions of the agents are marked with a dot and their corresponding numeral, while the small arrows at the end of the paths denote the final positions for the agents at the end of the time period of simulation. Figure 7.3(c) shows that the agents stay on the same line they started at as they converge to the saddle. The chattering that occurs at steady-state is due to the fixed integration step of the numerical solver.

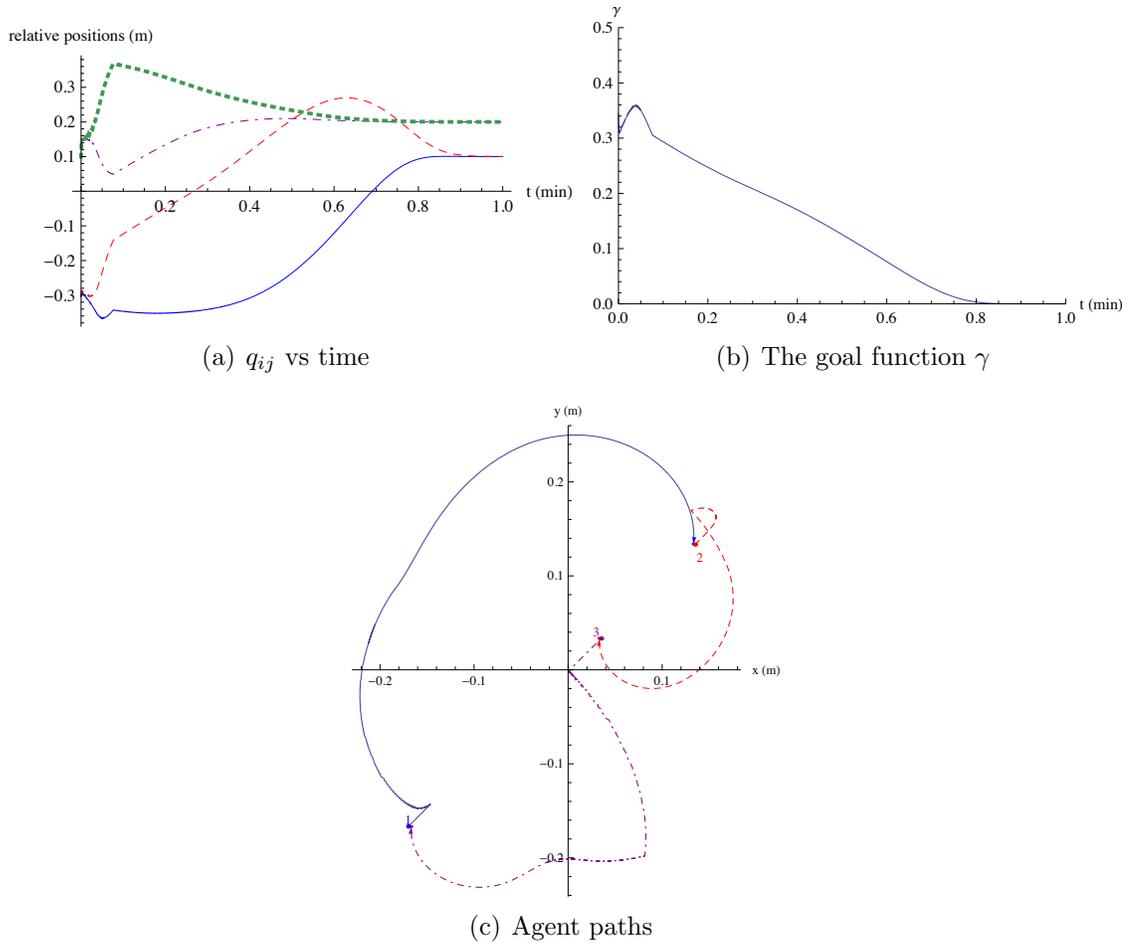


Figure 7.4: Simulation of a formation control maneuver where all agents have to switch positions, starting *close* but not in the attraction region of a saddle. The notation is the same as that in Figure 7.3. In the initial configuration, $q_{12}(0) = (-0.305, -0.3)$ m and $q_{23}(0) = (0.1, 0.1)$ m. In Figure 7.4(a) the continuous curve shows the evolution of $x_1 - x_2$, the dashed curve shows that of $y_1 - y_2$, the dot-dashed curve depicts $x_2 - x_3$, and the thick dotted one gives $y_2 - y_3$. Figure 7.4(b) indicates that the desired formation is reached since the goal function γ converges to zero. Figure 7.4(c) shows the paths of the three agents as they switch positions to reconfigure themselves into the desired formation.

positions have a fixed off-set with respect to those of Fig. 7.4. This goes to show that what is being controlled is the *relative* position of the agents, rather than their absolute position in some fixed coordinate frame.

In the second simulation test, we assess the system with six agents to establish its non-trivial nature. Along similar lines to the previous simulation, we set the initial and final configurations in a 45° straight line, in a way that they switch their positions in a manner shown in Figure 7.6. Figure 7.7 shows the simulation results.

The numerical simulations were performed on a 32 bit Intel Core 2 Duo processor (each core @ 2GHz) employed with a 3GB RAM. MATLAB software was used for the computation. Under such a computing environment, the times taken by different number of agents is shown in Table 7.1.

Table 7.1: Computation Time

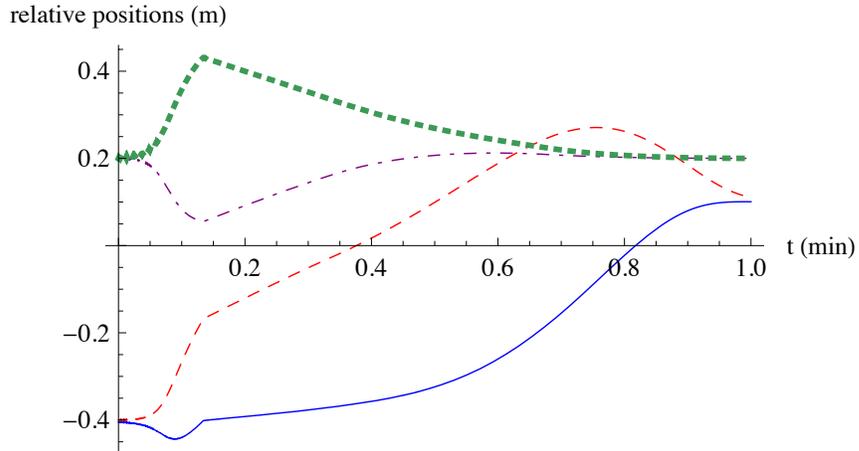
Agents	Time(sec)
3	28.735
4	55.446
5	93.620
6	320.235

7.2 Experimental Validation

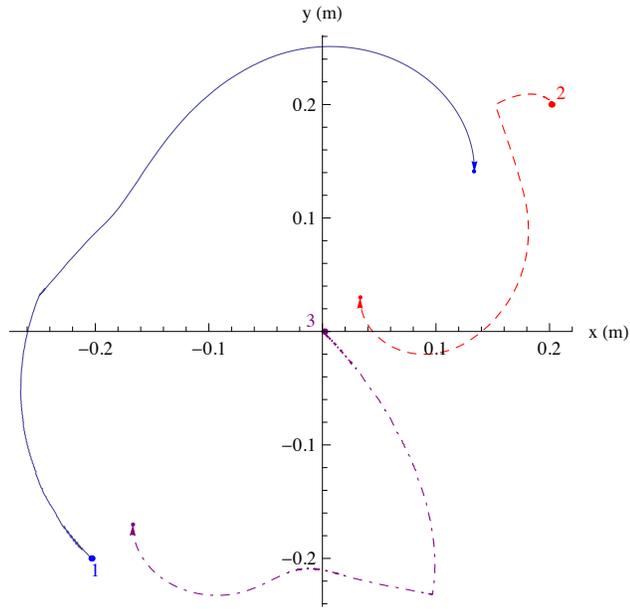
In this section we describe the hardware setup for the experimental demonstration of the formation control and present the related results.

7.2.1 Corobots

The Corobots, developed by the *Coroware Inc.*, are employed to act as mobile agents in our experiments. To each Corobot, a set of reflective markers is attached which enables the motion capture system to determine its location. The Corobots



(a) q_{ij} vs time



(b) Agent paths

Figure 7.5: Simulation of a formation control maneuver where all agents have to switch positions, starting in the neighborhood of a saddle configuration. The notation is the same as that in Fig. 7.3. The initial configuration, $q_{12}(0) = (-0.405, -0.4)$ m and $q_{23}(0) = (0.2, 0.2)$ m. In Fig. 7.5(a) shows with continuous line the evolution of $x_1 - x_2$, with dashed curve that of $y_1 - y_2$, with dot-dashed curve that of $x_2 - x_3$, and with thick dotted one the trajectory of $y_2 - y_3$. Figure 7.5(b) shows the paths of the three agents as they switch positions to reconfigure themselves into the desired formation, where it can be seen that their final position is different from that of the case depicted in Fig. 7.4.

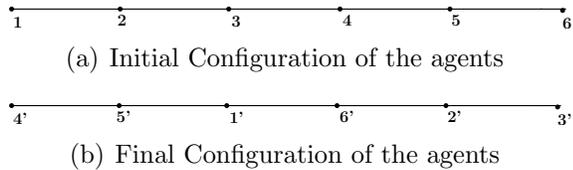


Figure 7.6: Initial and final configurations for the 6-agent simulation test

are programmed using the Player/Stage interface which works on the C/C++ platform.

The Corobots have a 1.5GHz processor, 1 GB of memory and an Ethernet card used for wireless communication. Each Corobot can hence be viewed as a computer capable of on board processing. They are also equipped with a 2 MP camera, encoders, laser ranger and bumper sensors. A 4-dof arm with a gripper sensor embedded to the Corobot platform allows some interaction with the workspace.

7.2.2 Vicor Motion Capture System

The Vicor motion capture system (Vicor MX in short) is used to obtain the real-time position information about the agent.

The Vicor system consists of 8 infrared cameras, each of them placed at certain height from the laboratory workspace. The cameras track the location of the markers embedded on the corobots which gives their current position. Though, mathematically, 3 cameras are sufficient to accurately determine the location of a marker in space, additional cameras are installed to exclude the possibility of 2 or more cameras getting obstructed (people moving around, desks obstructing, etc).

A C++ program is written to obtain the position of markers and the angular orientation of the axes on the robots. The numerical data so obtained from the Vicor form the input to the formation controller described in the next section.

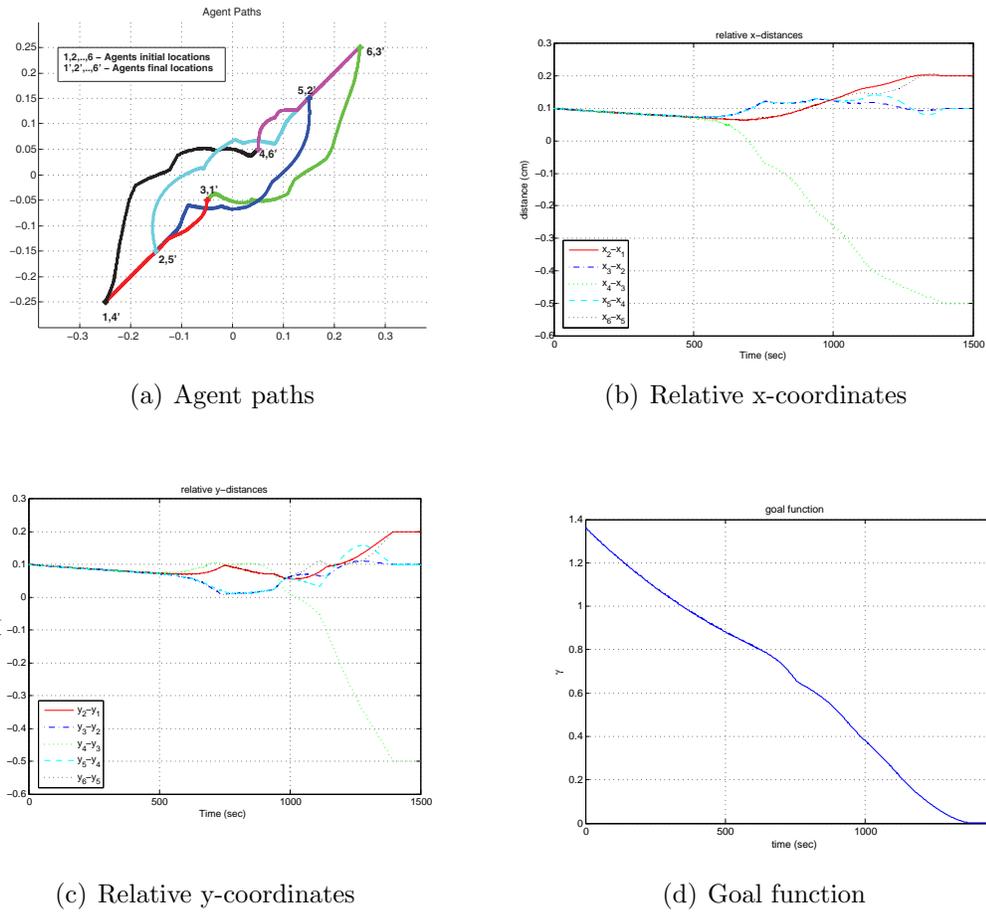


Figure 7.7: Simulation of a 6-agent formation control maneuver where the agents switch positions in accordance with Fig. 7.6. The agents begin on a 45° straight line in the neighborhood of a saddle configuration. The relative distances between successive agents is $(0.1, 0.1)$. Figure 7.7(a) shows the agent paths, with unprimed numbers representing the initial configuration and primed numbers the final configuration. Figure 7.7(b) shows the evolution of the relative x-distances between the agents. Figure 7.7(c) shows the evolution of the relative-ydistances between the agents.

7.2.3 The formation controller

The formation controller described here generates the linear velocity and angular velocity vectors that the Corobots take as input.

The gradient of the navigation function prescribes the desired relative velocity vectors between the agents. The relative velocities are then mapped to absolute velocities using the Moore-Penrose inverse. The inverse generates a least-norm solution to the absolute-velocities. The velocities so generated could be very high in terms of the magnitude due to the nature of the gradient of φ and may not be practically implementable on the Corobots. These velocities are therefore proportionally scaled down for all the agents to velocities attainable by the Corobots using the hyperbolic tangent function. The absolute velocity vectors obtained from the Moore-Penrose generalized inverse are then converted into velocity and angular velocity inputs to the Corobots using the following procedure.

Let f_x and f_y denote the absolute velocity vectors in x and y directions. The resultant direction of motion for each agent can be calculated using the inverse tangent of $\frac{f_y}{f_x}$. We denote the desired direction of motion θ_d .

The current direction of motion, θ_c , of the agent is obtained from the Vicon motion capture system. The error in θ is the difference between θ_c and θ_d *i.e.*,

$$e_\theta = \theta_c - \theta_d$$

We build a proportional controller for stabilizing the error e_θ

$$\dot{e}_\theta = -K(e_\theta) \tag{7.1}$$

Since $e_\theta = \theta - \theta_d$, we have

$$\dot{e}_\theta = \dot{\theta}_c - \dot{\theta}_d$$

since $\dot{\theta}_c = \omega$, we get

$$\omega = \dot{e}_\theta + \dot{\theta}_d \tag{7.2}$$

Substituting (7.1) in (7.2), we get,

$$\omega = -K_1 e_\theta + \dot{\theta}_d$$

At any position of the agent, the desired direction θ_d can be computed using the velocity gradients in the x and y directions as

$$\theta_d = \text{atan2}(f_y, f_x)$$

The angular velocity ω thus becomes

$$\omega = -K_1 e_\theta + \frac{d}{dt} \text{atan2}(f_y, f_x)$$

The above equation, on evaluating the derivative, becomes

$$\omega = -K_1 [\theta - \text{atan2}(f_y, f_x)] + \frac{v}{\|F\|^2} \left[f_x \left(\frac{\partial f_y}{\partial y} \sin \theta + \frac{\partial f_y}{\partial x} \cos \theta \right) - f_y \left(\frac{\partial f_x}{\partial y} \sin \theta + \frac{\partial f_x}{\partial x} \cos \theta \right) \right]$$

The velocity vector as described earlier is given by

$$v = K_2 \tanh(f)$$

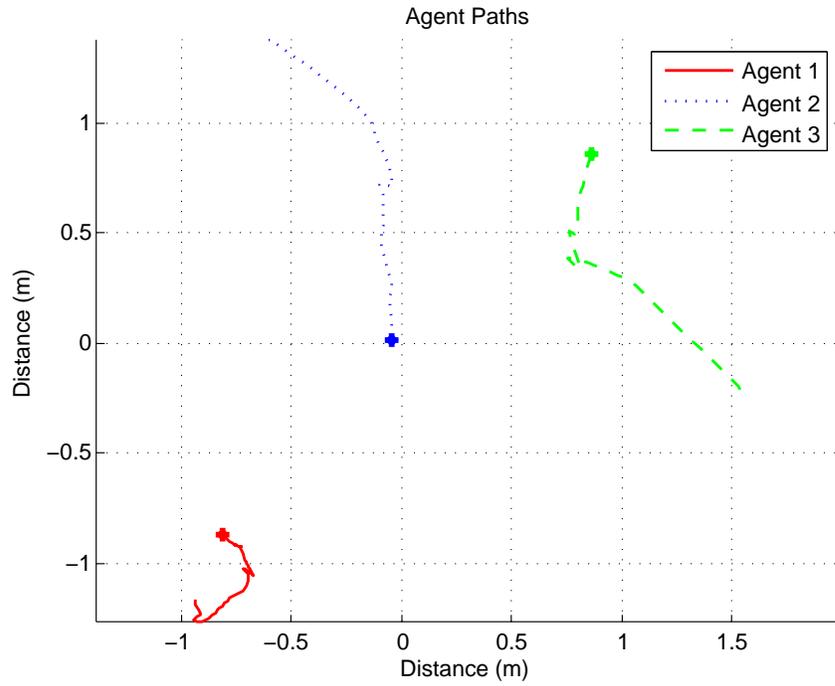
where f is the absolute velocity vector.

7.2.4 Experimental results

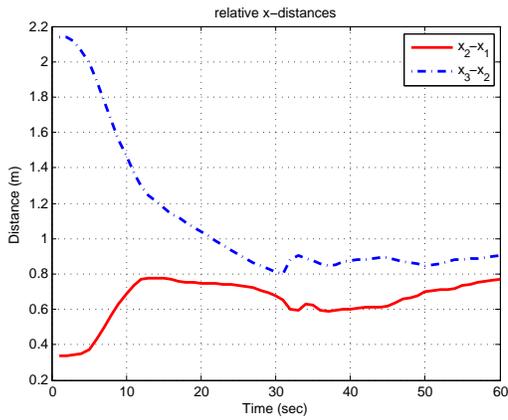
We present the results from our laboratory, with the experiments performed on the Corobots. We employed 3 Corobots to analyze the behavior the system. Three tests were performed to make the analysis exhaustive. In the first case, the agents were to form a straight line from almost any initial configuration, the results of which are shown in Figure 7.8. In the second case, the agents switch their relative positions in a straight line along a 45° line. The test results are shown in Figure 7.9. In the final case, the agents switch their positions in a triangle. Figure 7.10 shows the results. Again, we see from Figures 7.9 and 7.10 that it is the relative positions among the agents that is being switched and not their absolute positions. The dot signifies the destination location of the agent.

The jitter that we see in the agent paths is due to the possible overshoot of the Corobots occurring due to the fixed time step associated with each Corobot.

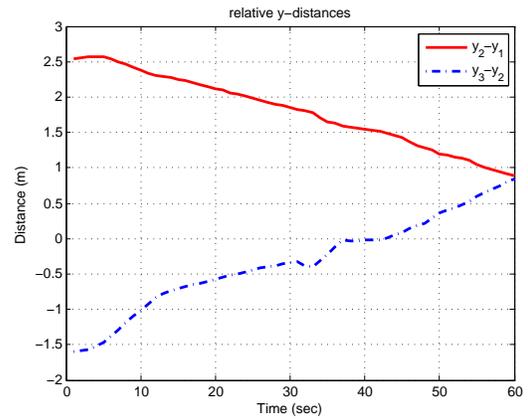
Close to the destination, the gradient vector varies sharply and hence difficult to be followed by Corobots. The feedback obtained from the Vicon maintains the Corobot close to the required trajectory. This jitter might sometimes be useful, in that, the configuration could escape from any possible saddle points at which the Corobots might get stuck.



(a) Agent Paths

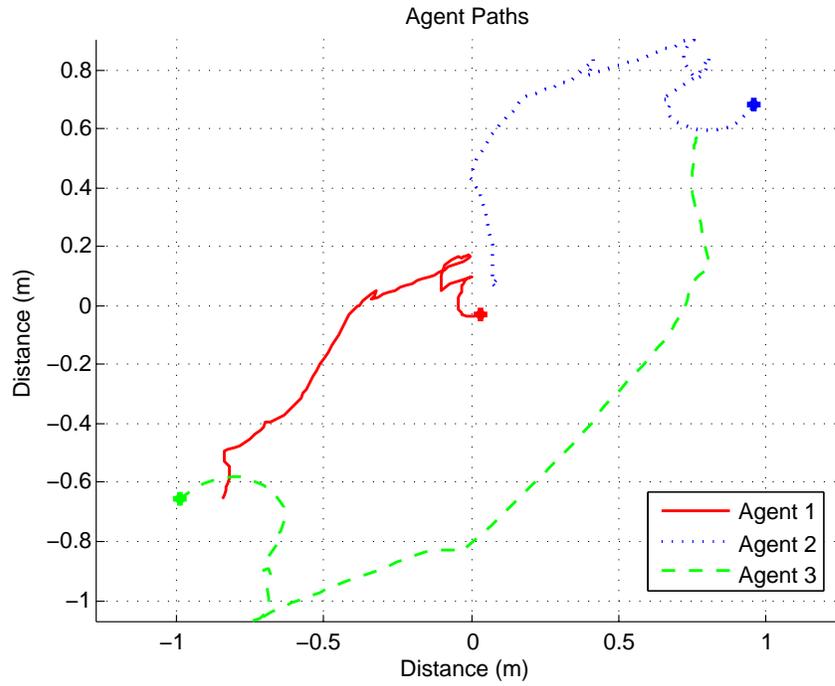


(b) relative x-distances

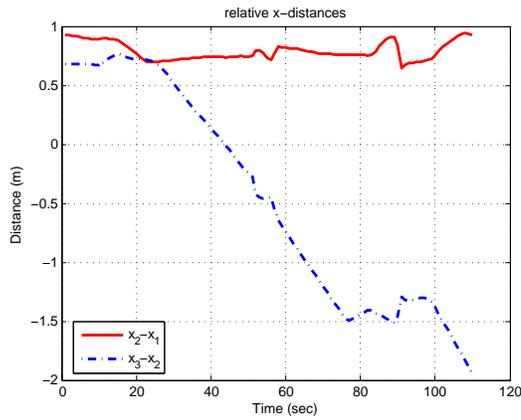


(c) relative y-distances

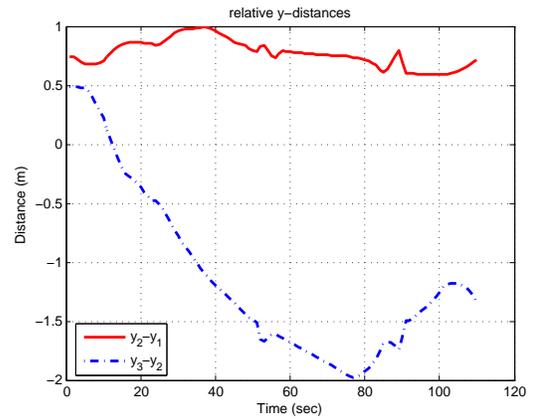
Figure 7.8: Experimental results for the line formation of three agents from almost any initial configuration



(a) Agent Paths

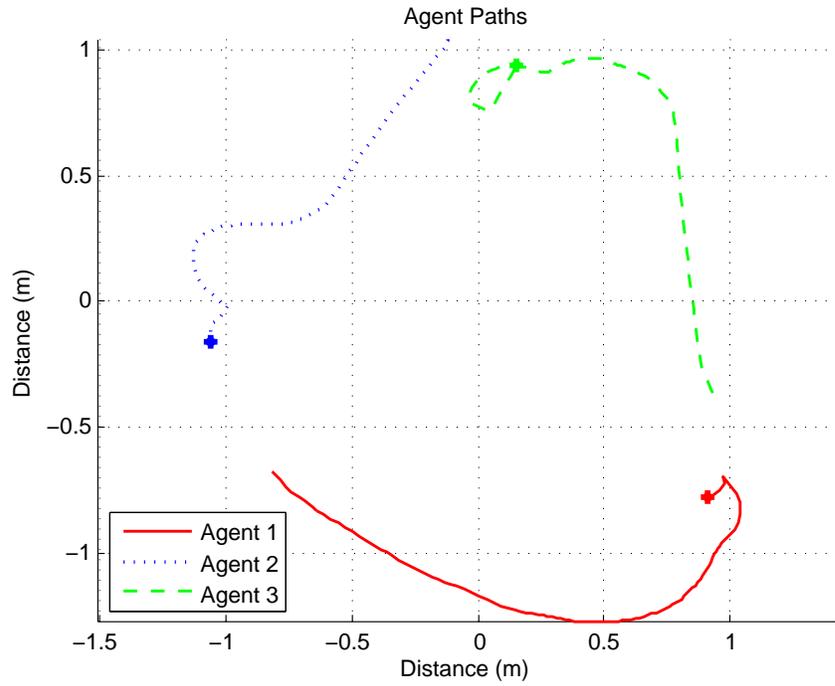


(b) relative x-distances

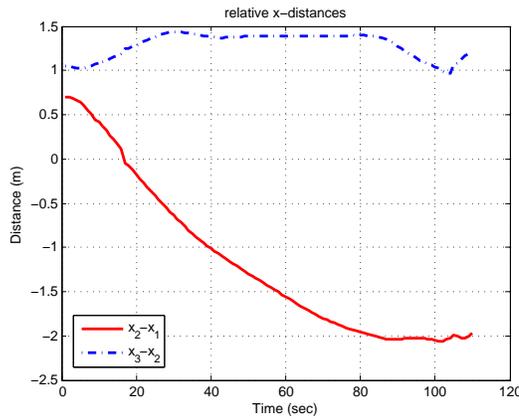


(c) relative y-distances

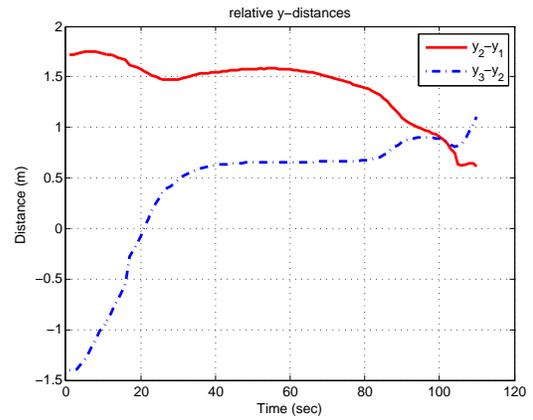
Figure 7.9: Experimental results showing the switching of relative positions of three agents in a straight line.



(a) Agent Paths



(b) relative x-distances



(c) relative y-distances

Figure 7.10: Experimental results showing three agents switching in a triangle.

Chapter 8

CONCLUSIONS AND FUTURE WORK

This work presents a methodology to drive a group of robots from an arbitrary initial configuration to a free-floating destination formation. This was an open problem because the generalization of the single agent navigation function to multiple agents is problematic. The approach adopted in the literature in the construction of the navigation for multiple agents employs the product of positive semidefinite scalar functions as a metric of distance of the system from collision configurations. We indicate that in that context, and following the proof techniques that have appeared in literature, one may fail to establish the Morse character of the functions critical points, on which the convergence properties of the potential field that function generates, rely upon.

In this work, we propose an alternative construction of the potential function for which we can establish the properties sufficient for the (almost global) convergence of the system to the desired configuration. The new construction is a nonsmooth positive definite function, and using the proof techniques based on nonsmooth analysis, and control theory for dynamical systems expressed in the form of finite dimensional differential inclusions we establish the convergence properties of the multi-agent system with single integrator kinematics.

A centralized planning scheme has been employed in the construction of the new navigation function. Given the limitations of the centralized architecture, this work could be extended to achieve the formation control of multiple robots using the

decentralized control scheme. The construction of the obstacle function, β , in our navigation function facilitates decentralization. Assuming each agent has knowledge about the agents only in its neighborhood, a local obstacle function, say β_i , for each agent i can be constructed with agents within that neighborhood. Similarly, a goal function γ_{d_i} , is constructed for the same neighborhood. A local navigation function φ_i , when minimized for agent i would steer it towards the destination formation.

The agents considered in the multi-agent system are assumed to be holonomic agents with omnidirectional properties. In our implementation, we employed control designs [46] that allowed the application to nonholonomic systems, but vehicle orientation is not controlled. Another important issue with the multi-agent navigation functions is the parameter tuning process. So far, no formal procedure to tune the navigation function has been published. In this work, the tuning has been done using the trial and error approach. Developing a better technique for the parameter tuning, possibly through learning, can be taken up as a good challenge.

BIBLIOGRAPHY

- [1] A.F.Filippov. *Differential Equations with Discontinuous Righthand Sides*. Norwell, M.A:Kluwer, 1988.
- [2] A. A. Agrachev, D. Pallaschke, and S. Scholtes. On morse theory for piecewise smooth functions. *Journal of Dynamical and Control Systems*, 3:449–469, October 1997.
- [3] Yu Aminov. *Differential Geometry and Topology of Curves*. CRC Press, 2000.
- [4] Jean-Pierre Aubin. *Optima and Equilibria: An Introduction to Nonlinear Analysis*. Springer, 1998.
- [5] A. Bacciotti and F. Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth lyapunov functions. *ESAIM: Control Optimisation and Calculus of Variations*, 4:361–376, 1999.
- [6] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, December 1998.
- [7] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. "Robots in Unstructured Environments," 91 ICAR., Fifth International Conference on*, pages 1012 –1017 vol.2, June 1991.
- [8] J. Barraquand and J.-C. Latombe. A monte-carlo algorithm for path planning with many degrees of freedom. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1712 –1717 vol.3, May 1990.
- [9] J. Barraquand and J.C. Latombe. A distributed representation approach. *Int. Journal of Robotics Research*, 10(6):628–649, 1991.
- [10] Adi Ben-Israel and Thomas N.E. Greville. *Generalized inverses: Theory and applications*. Wiley, New York, 1974.

- [11] R.W. Brockett. On the computer control of movement. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 534–540 vol.1, apr 1988.
- [12] Howie Choset. Nonsmooth analysis, convex analysis, and their applications to motion planning. *International Journal of Computational Geometry and Applications*, pages 447–469.
- [13] Howie Choset. Nonsmooth analysis, convex analysis, and their applications to motion planning. *International Journal of Computational Geometry and Applications*, 9(4-5):447–469, 1999.
- [14] C.M. Clark, T. Bretl, and S. Rock. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 7, pages 7–3621 – 7–3631 vol.7, 2002.
- [15] C.M. Clark, S.M. Rock, and J.-C. Latombe. Motion planning for multiple mobile robots using dynamic networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 4222 – 4227 vol.3, sept 2003.
- [16] Frank H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.
- [17] D.C. Conner, A.A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3546 – 3551 vol.3, oct. 2003.
- [18] J. Cortes. Discontinuous dynamical systems. *Control Systems, IEEE*, 28(3):36–73, June 2008.
- [19] M.C. De Gennaro and A. Jadbabaie. Formation control for a cooperative multi-agent system using decentralized navigation functions. In *Proceedings of American Control Conference*, page 6 pp., 14-16 2006.
- [20] D.V. Dimarogonas, K.J. Kyriakopoulos, and D. Theodorakatos. Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2430–2435, 15-19 2006.
- [21] D.V. Dimarogonas, S.G. Loizou, K.J. Kyriakopoulos, and M.M. Zavlanos. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42(2):229–243, February 2006.

- [22] S. I. Dubov. On the generalized gradient of the distance function. *Journal of mathematical sciences*, 100(6):2593–2600, 2000.
- [23] M. Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, dec 2001.
- [24] Victor Guillemin and Alan Pollack. *Differential Topology*. Prentice-Hall, 1974.
- [25] H.Th Jongen and D. Pallaschke. On linearization and continuous selections of functions. *Optimization*, 19(3):343–353, 1988.
- [26] V. Kallem, A.T. Komoroski, and V. Kumar. Sequential composition for navigating a nonholonomic cart in the presence of obstacles. *IEEE Transactions on Robotics*, 27(6):1152–1159, dec. 2011.
- [27] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 500 – 505, March 1985.
- [28] D. E. Koditschek and E. Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442, 1990.
- [29] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [30] N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2968 –2973 vol.3, 2001.
- [31] Savvas G. Loizou and Kostas J. Kyriakopoulos. Closed loop navigation for multiple holonomic vehicles. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2861–2866, 2002.
- [32] S.G. Loizou and K.J. Kyriakopoulos. Navigation of multiple kinematically constrained robots. *IEEE Transactions on Robotics*, 24(1):221–231, February 2008.
- [33] V. Manikonda, P.S. Krishnaprasad, and J. Hendler. A motion description language and a hybrid architecture for motion planning with nonholonomic robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 2021 –2028 vol.2, may 1995.
- [34] B. Mendelson. *Introduction to Topology*. Boston:Allyn and Bacon, 1975.
- [35] John Milnor. Morse theory. *Annals of Mathematics Studies*, 1963.

- [36] P. Ogren and N.E. Leonard. Obstacle avoidance in formation. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 2492 – 2497 vol.2, 14-19 2003.
- [37] Reza Olfati-saber and Richard M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *in IFAC World Congress*, 2002.
- [38] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.
- [39] E. P. Ryan. An integral invariance principle for differential inclusions with applications in adaptive control. *SIAM J. Control Optim*, 36:960–980, 1998.
- [40] R. O. Saber, W. B. Dunbar, and R. M. Murray. Cooperative control of multi-vehicle systems using cost graphs and optimization. In *Proceedings of the American Control Conference, Denver, Colorado*, volume 3, pages 2217–2222, June 2003.
- [41] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*. Publish or Perish Inc., 1999.
- [42] Christopher Clark Stephen and Stephen Rock. Randomized motion planning for groups of nonholonomic robots. In *International Symposium of Artificial Intelligence, Robotics and Automation in Space*, 2001.
- [43] Herbert G. Tanner and Amit Kumar. Formation stabilization of multiple agents using decentralized navigation functions. In S. Thrun, G. Sukhatme, S. Schall, and O. Brock, editors, *Robotics: Science and Systems I*, pages 49–56. Springer, 2005.
- [44] Herbert G. Tanner and Amit Kumar. Towards decentralization of multi-robot navigation functions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4143–4148, Barcelona, Spain, 2005. IEEE.
- [45] Herbert G. Tanner, Savvas G. Loizou, and Kostas J. Kyriakopoulos. Non-holonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation*, 19:53–64, 2003.
- [46] H.G. Tanner and K.J. Kyriakopoulos. Nonholonomic motion planning for mobile manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1233–1238 vol.2, 2000.

- [47] H.G. Tanner, G.J. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443 – 455, June 2004.
- [48] C.W. Warren. Multiple robot path coordination using artificial potential fields. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 500 –505 vol.1, May 1990.
- [49] L.L. Whitcomb, D.E. Koditschek, and J.B.D. Cabrera. Toward the automatic control of robot assembly tasks via potential functions: the case of 2-d sphere assemblies. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2186 –2191 vol.3, May 1992.
- [50] Ya. Izrailavich T. Fomenko Yu. Borisovich, N. Bliznyakov. *Introduction to Topology*. Mir Publishers, 1985.
- [51] M. M. Zavlanos and K. J. Kyriakopoulos. Decentralized motion control of multiple mobile agents. In *Proceedings of the 11th Mediterranean Conference on Control and Automation, Rhodes, Greece, 2003*.
- [52] M. M. Zavlanos and K. J. Kyriakopoulos. A distributed representation approach motion control of multiple mobile agents. In *Proceedings of the 11th Mediterranean Conference on Control and Automation, Rhodes, Greece, 2003*.

Appendix

DEFINITIONS

Definition 17 (Homeomorphism [50]). *A mapping $f : X \rightarrow Y$ of metric spaces is called a homeomorphism, and the spaces X, Y homeomorphic if (1) f is bijective, (2) f is continuous, and (3) the inverse mapping f^{-1} is continuous.*

Definition 18 (Diffeomorphism [24]). *A smooth map $f : X \rightarrow Y$, of subsets of two Euclidean spaces is a diffeomorphism if it is one to one and onto, and if the inverse map $f^{-1} : Y \rightarrow X$ is also smooth. X and Y are diffeomorphic if such a map exists.*

Definition 19 (Limit Point [50]). *A point $x \in \mathbb{A}$ is said to be a limit point of the set \mathbb{A} if there is atleast one point $x' \in \mathbb{A}$ other than x in each neighborhood $\Omega(x)$ of the point x .*

Definition 20 (Closed Set [50]). *Let \mathcal{X} be a topological space. A set $\mathbb{A} \subset \mathcal{X}$ is closed if and only if it contains all its limit points.*

Definition 21 (Connected Space [34]). *A topological space \mathcal{X} is said to be connected if the only two subsets of \mathcal{X} that are simultaneously open and closed are \mathcal{X} itself and the empty set Φ .*

Definition 22 (Analytic function [41]). *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is analytic at $a \in \mathbb{R}^n$ if f can be expressed as a power-series in the $(x^i - a^i)$ which converges in some neighborhood of a .*

Definition 23 (Manifold [41]). *A manifold is a metric space M with the following property:*

If $x \in M$, then there is some neighborhood U of x and some integer $n \geq 0$ such that U is homeomorphic to \mathbb{R}^n . An n -dimensional manifold thus is “locally” similar to the metric space \mathbb{R}^n .

Definition 24 (Convex Hull[12]). *The convex hull of a set of vectors $\{v_i : i = 1, \dots, n\}$ is*

$$\text{Co} \{v_i : i = 1, \dots, n\} = \{\sum_{i=1}^n \lambda_i v_i : \lambda_i \in \mathbb{R} \text{ such that } \lambda_i > 0 \forall i \text{ and } \sum_{i=1}^n \lambda_i = 1\}$$

Definition 25 (Continuous selection of functions [2]). *Let \mathcal{M} be an n -dimensional manifold topological, and let $f : M \rightarrow \mathbb{R}$ and $f_1, \dots, f_m : M \rightarrow \mathbb{R}$ be continuous functions. If $I(x) = \{i \in \{1, \dots, m\} | f_i(x) = f(x)\}$ is nonempty at every point $x \in M$, then f is called a continuous selection of functions f_1, f_2, \dots, f_m . We denote by $CS(f_1, \dots, f_m)$ the set of all continuous selections of f_1, f_2, \dots, f_m . The set $I(x)$ is called the active index set of f at the point x .*

Definition 26 (Nonsmooth critical point [25]). *If \bar{x} is a critical point for $f \in CS(f_i, i \in I)$, then there exist real numbers $\lambda_i, i \in I(\bar{x})$ with*

$$\sum_{i \in I(\bar{x})} \lambda_i df_i(\bar{x}) = 0, \sum_{i \in I(\bar{x})} \lambda_i = 1, \lambda_i \geq 0, i \in I(\bar{x})$$

Definition 27 (Nondegenerate critical point-The nonsmooth case [2]). *Let M be a smooth n -dimensional manifold, $f_1, f_2, \dots, f_m : M \rightarrow \mathbb{R}$ be C^2 -functions, and $f \in CS(f_1, f_2, \dots, f_m)$. A critical point $x_0 \in M$ of f is called nondegenerate if the following two conditions hold:*

(ND1) *For each $i \in \hat{I}(x_0)$, the set of differentials $\{df_j(x_0) | j \in \hat{I}(x_0) \setminus \{i\}\}$ is linearly independent;*

(ND2) *The second differential $d^2L(., \hat{\lambda})(x_0)$ of $x \rightarrow L(x, \hat{\lambda})$ is regular on*

$$\hat{T}(x_0) = \bigcap_{i \in \hat{I}(x_0)} \text{kern } df_i(x_0), \quad (\text{A.1})$$

where,

$$L(x, \lambda) = \sum_{i \in \hat{I}(x_0)} \lambda_i f_i(x),$$

and the reals λ_i are such that

$$dL(\cdot, \hat{\lambda})(x_0) = 0, \quad \sum_{i \in \hat{I}(x_0)} \lambda_i = 1, \quad \hat{\lambda}_i \geq 0 \forall i \in \hat{I}(x_0)$$

The linear subspace $\hat{T}(x_0)$ at x_0 is nothing but the tangent space at x_0 and can be written as

$$\hat{T}(x_0) = \left\{ \zeta \in \mathbb{R}^n : \zeta^T df_i(x_0) = 0 \forall i \in \hat{I} \right\}$$

and for the second condition ND2 to be satisfied the following condition needs to be satisfied

$$\nu^T (d^2L(\cdot, \hat{\lambda})(x_0)) \nu \neq 0$$

where ν is a column vector in the tangent space $\hat{T}(x_0)$.

Definition 28 (Continuous function [34]). Let (X, d) and (Y, d') be metric spaces, and let $a \in X$. A function $f : X \rightarrow Y$ is said to be continuous at the point $a \in X$ if given $\epsilon > 0$, there is a $\delta > 0$, such that

$$d'(f(x), f(a)) > \epsilon$$

whenever $x \in X$ and

$$d(x, a) > \delta$$

The function $f : X \rightarrow Y$ is said to be continuous if it is continuous at each point of X .

Definition 29 (Isolated Point [50]). A point $x \in A$ is said to be isolated if there is a neighborhood $\Omega(x)$ of the point x such that it does not contain any points of the set A other than x .

Definition 30 (Upper semicontinuous function [18]). The function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is upper semicontinuous at $x \in \mathbb{R}^d$ if $-f$ is lower semicontinuous at x .

Definition 31 (Lower semicontinuous function [18]). A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is lower semicontinuous at $x \in \mathbb{R}^d$ if, for all $\epsilon \in (0, \infty)$, there exists $\delta \in (0, \infty)$ such that, for $y \in B(x, \delta)$, $f(y) \geq f(x) - \epsilon$. The function f is lower semicontinuous if and only if its epigraph is closed.

Definition 32 (Epigraph [18]). The epigraph of a function $f(x)$ is the set of points lying on or above its graph, that is

$$\text{epi}(f) = \{(x, \mu) \in \mathbb{R}^d \times \mathbb{R} : f(x) \leq \mu\} \subset \mathbb{R}^{d+1}$$

Definition 33 (Curve [3]). A set of the Euclidean space is called an elementary curve if this set is the image of an interval of the real axis under a one-to-one continuous map, whose inverse map is continuous too.

Definition 34 (Manifold-with-boundary [41]). A manifold-with-boundary is a metric space M with the following property:

If $x \in M$, then there is some neighborhood U of x and some integer $n \geq 0$ such that U is homeomorphic to either \mathbb{R}^n or \mathbb{H}^n (the closed half-space).

Definition 35 (Convex function [12]). A function f is said to be convex if $\forall x_i$

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i), i = 1, \dots, n \sum_{i=1}^n \lambda_i = 1 \text{ and } 0 \leq \lambda_i \leq 1$$

Definition 36 (Absolutely continuous function [18]). A function $\gamma : [a, b] \rightarrow \mathbb{R}$ is absolutely continuous if, for all $\epsilon \in (0, \infty)$, there exists $\delta \in (0, \infty)$ such that, for

each finite collection $\{(a_1, b_1), \dots, (a_n, b_n)\}$ of disjoint open intervals contained in $[a, b]$ with $\sum_{i=1}^n (b_i - a_i) < \delta$, it follows that

$$\sum_{i=1}^n \|\gamma(b_i) - \gamma(a_i)\| < \epsilon$$

Definition 37 (Moore Penrose Generalized inverse [10]). *The Moore-Penrose generalized inverse or pseudoinverse of a finite matrix A (square or rectangular) of real or complex elements is a matrix X which satisfies the following equations (the Penrose equations)*

- $AXA = A$
- $XAX = X$
- $(AX)^* = AAX$
- $(XA)^* = XA$

where $()^*$ denotes the conjugate transpose of the matrix.

For a real matrix A , the pseudoinverse, written as A^\dagger is

$$A^\dagger = (A^T A)^{-1} A^T$$

Definition 38 (Continuously differentiable function [4]). *Let $f : X \rightarrow \mathbb{R}$ be a function with non-empty domain. f is said to be continuously differentiable in its domain if the function $\nabla f(x)$ is continuous, where $\nabla f(x)$ is the gradient of f at x .*

Definition 39 (Lower semicontinuous set-valued map [18]). *A set-valued map $F : \mathbb{R}^d \rightarrow \mathcal{B}(\mathbb{R}^d)$ is lower-semicontinuous at $x \in \mathbb{R}^d$, if for all $\epsilon \in (0, \infty)$, there exists $\delta \in (0, \infty)$ such that $F(x) \subseteq F(y) + B(0, \epsilon)$ for all $y \in B(x, \delta)$.*

Definition 40 (Upper semicontinuous set-valued map [18]). *A set-valued map $F : \mathbb{R}^d \rightarrow \mathcal{B}(\mathbb{R}^d)$ is upper-semicontinuous at $x \in \mathbb{R}^d$, if for all $\epsilon \in (0, \infty)$, there exists $\delta \in (0, \infty)$ such that $F(y) \subseteq F(x) + B(0, \epsilon)$ for all $y \in B(x, \delta)$.*

Definition 41 (Critical point [28]). *If $h : E^n \rightarrow E^m$ then Dh denotes the Jacobian—that is, the matrix of partial derivatives of h . If $[Dh](x)$ is not surjective then $x \in E^n$ is a critical point of h .*

Definition 42 (Non-degenerate critical point [28]). *Let $\varphi \in C^2[E^n, E]$. A critical point of φ is non-degenerate if the Hessian, $D^2\varphi$, has full rank at that point.*

Definition 43 (Saddle point [28]). *Any non-degenerate critical point which is neither a maximum nor a minimum is called a saddle point.*

Definition 44 (Morse index of a critical point [28]). *The Morse index of $f(x)$ at a critical point x_0 is the dimension of the subspace of \mathbb{R}^n spanned by eigenvectors of the Hessian.*

Proposition 14. *Let x_1, x_2, \dots, x_n belong to a linearly independent set $\mathcal{X} \subset \mathbb{R}^n$, and $y \in \mathbb{R}^n$ be added to each of the vectors x_1, x_2, \dots, x_n in \mathcal{X} . If y can be written as a linear combination of the vectors in \mathcal{X} , then the condition for the vectors $y + x_1, y + x_2, \dots, y + x_n$ to be linearly independent is that $\mu_1 + \mu_2 + \dots + \mu_n = -1$, where $\mu_1, \mu_2, \dots, \mu_n$ are the coefficients in the linear combination of the vectors $y + x_1, y + x_2, \dots, y + x_n$*

Proof. Since the vectors x_1, x_2, \dots, x_n are linearly independent, we have

$$\lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n = 0 \Leftrightarrow \lambda_1 = \lambda_2 = \dots = \lambda_n = 0$$

Let y be expressed as a linear combination of the vectors $x_1, x_2, \dots, x_n \in \mathcal{X}$ with $\mu_1, \mu_2, \dots, \mu_n$ as respective coefficients, *i.e.*

$$y = \mu_1 x_1 + \mu_2 x_2 + \dots + \mu_n x_n$$

If y is added to the set \mathcal{X} , then the new vectors obtained are

$$y + x_1, y + x_2, \dots, y + x_n$$

Assume $\lambda_1 (y + x_1) + \lambda_2 (y + x_2) + \dots + \lambda_n (y + x_n) = 0$ with $|\lambda_1| + |\lambda_2| + \dots + |\lambda_n| \neq 0$
 substituting for y in the above equation we get,

$$\lambda_1 (\mu_1 x_1 + \dots \mu_n x_n + x_1) + \lambda_2 (\mu_1 x_1 + \dots \mu_n x_n + x_2) + \dots + \lambda_n (\mu_1 x_1 + \dots \mu_n x_n + x_n) = 0$$

rearranging the terms in above equation we get

$$x_1 [(\lambda_1 + \dots \lambda_n) \mu_1 + \lambda_1] + x_2 [(\lambda_1 + \dots \lambda_n) \mu_2 + \lambda_2] + \dots + x_n [(\lambda_1 + \dots \lambda_n) \mu_n + \lambda_n] = 0$$

Since \mathcal{X} is a linearly independent set none of its elements x_1, x_2, \dots, x_n is a zero vector. Thus we have

$$\begin{aligned} (\lambda_1 + \dots \lambda_n) \mu_1 + \lambda_1 &= 0 \\ (\lambda_1 + \dots \lambda_n) \mu_2 + \lambda_2 &= 0 \\ &\vdots \\ (\lambda_1 + \dots \lambda_n) \mu_n + \lambda_n &= 0 \end{aligned}$$

Rearranging the above equations, we get

$$\begin{aligned} (\mu_1 + 1)\lambda_1 + \lambda_2\mu_1 + \lambda_3\mu_1 \dots + \lambda_n\mu_1 &= 0 \\ \mu_2\lambda_1 + (\mu_2 + 1)\lambda_2 + \lambda_3\mu_2 + \dots + \lambda_n\mu_2 &= 0 \\ &\vdots \\ \mu_n\lambda_1 + \lambda_2\mu_n + \lambda_3\mu_n \dots + \lambda_n(\mu_n + 1) &= 0 \end{aligned}$$

when written in matrix notation, we have

$$\begin{bmatrix} \mu_1 + 1 & \mu_1 & \dots & \mu_1 \\ \mu_2 & \mu_2 + 1 & \dots & \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_n & \dots & \mu_n + 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = 0$$

Now, since $|\lambda_1| + |\lambda_2| + \dots + |\lambda_n| \neq 0$, equation (A) will have solutions for $\det(A) = 0$, where

$$A = \begin{bmatrix} \mu_1 + 1 & \mu_1 & \dots & \mu_1 \\ \mu_2 & \mu_2 + 1 & \dots & \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_n & \dots & \mu_n + 1 \end{bmatrix}$$

The above determinant can be calculated as

$$\mu_1 + \mu_2 + \mu_3 + \dots + \mu_n + 1$$

which proves the proposition. □