FLOCKING WITH FORMATION CONTROL IN MOBILE SENSOR NETWORKS FOR AREA SEARCH

by

Luis Ariel Valbuena Reyes

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

Fall 2012

FLOCKING WITH FORMATION CONTROL IN MOBILE SENSOR NETWORKS FOR AREA SEARCH

by

Luis Ariel Valbuena Reyes

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

Approved: _____
Suresh G. Advani, Ph.D.
Chair of the Department of Mechanical Engineering

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

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

ACKNOWLEDGMENTS

Special thanks:

To my brother, that special person who is always with me in all paths and journeys.

To my father, my mother and sister, the ones that truly love me.

To Dr. Herbert Tanner, for his infinite patience and advice.

To all people working in the cooperative robotics laboratory at University of Delaware.

To the grants NSF # 0447898 and NSF # 0822845, for the financial support.

TABLE OF CONTENTS

Appendix

LIST OF FIGURES

ABSTRACT

This thesis is addressing the integration of path following and formation control of a group of mobile robots to achieve flocking along specific path in the workspace. Several robots with similar capabilities are initially located in random positions and then move into a specified formation while performing path following over a designed route. The robots have to align their velocities to a common vector while approaching constant relative positions asymptotically.

This problem has practical significance because it is the basic component for the development of a mobile sensor network that searches for and harvests information. Agent cooperation and coordination is motivated by several examples from nature where animals increase their chances of survival and gathering food by working in groups. Robot groups can apply the same idea to improve their coverage and reduce mission time. Wider coverage and versatility are some of the features that a (wireless) sensor network acquires when enabled to move, making it useful for surveillance and environmental inspection.

Existing work has covered several aspects of the problem considered but it often ignores or tries to circumvent the discontinuities introduced into the system dynamics by the switching phenomena due to the operation of sensors and range of the communication. Another issue is the simplified dynamics (or kinematics) considered in cooperative control approaches, which raises some questions regarding its suitability for implementation on real platforms. Finally, although there is some limited work that combines formation control with flocking [46], these and other mission objectives - such as path following - may interfere with each other in existing formulations causing unpredictable behavior.

To address these technical challenges we integrate various control designs into a coherent policy with guaranteed performance properties. The first of these components is a formation control via a nonsmooth artificial potential function that is in charge of making the robots fall into a formation specification while avoiding collisions among the robots. The second component is a flocking control law in concert with formation stabilization. A third component is path following to drive the formation along the designed route (does not require switching among the primitives that compose the route). To develop this methodology we had to make use of mathematical tools from nonlinear control and nonsmooth analysis that enable us to study both the convergence of our proposed control law over a nonlinear nonsmooth nonholonomic dynamic system.

Chapter 1

INTRODUCTION

In this thesis, we present the integration of a formation controller with flocking control law and a path following algorithm in such a way these all behaviors work simultaneously in an orchestrated manner. Flocking and schooling behavior is observed in groups of animals such as cranes, antelopes and sardines, coordinated motion pattern performed without any explicit communication among the agents involved. [1] In our robotic implementation, the agents are mobile robots with differential steering. The goal is to search an area acting as a composite big sensor array distributed around an area we want to search. To achieve this behavior, we coordinate the robots so that they align their velocities to a common vector (that has the same direction of the path) while approaching constant relative position and follow a designated path.

We present a formal mathematical analysis that establishes the asymptotic convergence properties of the proposed methodology. We also present simulations and experiments that show that the method is realizable and our technical assumptions are reasonable.

1.1 Motivation

Over the last two decades, we gained access to better and more inexpensive processors, faster and more robust communication channels, more flexible manufacturing processes (3D printing) , more efficient techniques to store energy and plenty of choices for new materials. These new resources have fueled research on autonomous decision

[1] There is not a message passing protocol to make the animals in the flock agree on this behavior.

making machines that operate with decreased human intervention. Countries such as the United States, Israel and South Korea has increased their military investment towards robotics technology [4]. As autonomous systems are able to communicate to each other, the trend is to get simpler machines working cooperatively to outperform the work done by a single monolithic unit [25]. Tasks such as AHS (Automated Highway Systems), humanitarian demining, surveillance, satellite clustering and several other applications demand a group of mobile platforms to work together and coordinate their movements.

The advent of a extended variety of inexpensive and robust hardware makes easier the construction of mobile robots capable of doing calculations that are relatively demanding, allowing the implementation of results in nonlinear control design technology, and encouraging further research on multi-agent coordination.

The benefits of using multiple agents can be illustrated using the example of planting seeds in a field. An agent (a person or a robot) plants coffee seeds in a 24×24 squared meters field with an average velocity of 0.25 m/min so it takes around 2.3 days to do the job. The same proportion is achieved for tasks such as inspection, watering, fertilization and harvest. If two agents are doing the job on the same field it would take them 1.15 days, four agents would take 13.8 hours and so on. So for tasks like the one in the example that can be parallelized, the more agents you use to do the job the faster you can get it done.

Social insects provide a marvelous example on how an intelligent system can be created form a large number of individuals despite noise, errors in processing information and lack of global communication [47]. They are able to coordinate their actions to accomplish tasks that are beyond the capabilities of a single individual and particularly in the case of ants, the bigger their number, the more menacing they are. Extending this concept to the case of multiple robots and wireless sensor networks, gives rise to a flexible mobile sensor network, i.e, a sensor network that is easily scalable and can be deployed over a particular environment to take measurements. These networks can also find applications in manufacturing processes, and permit the more

efficient utilization of natural resources, [49].

1.2 Contributions

We present a control architecture that integrates path following and formation control plus flocking in a way that the control components responsible for these individual basic behaviors do not affect each other. It is common to model a system with multiple behaviors as a finite state machine, a Petri net or a hybrid system, that switches among individual behaviors one at a time; however, in our formulation the behaviors coexist and evolve simultaneously.

Our framework takes into account the nonholonomic nature of our robots by transforming their dynamic model into an input-output model, and deriving control laws for the resulting second order integrator dynamics. The approach taken to solve the formation control problem exploits some properties of nonsmooth functions such as min function, which identifies the cases in which a particular pair of robots are close to collision; however, the discontinuities introduced in the dynamics cannot be analyzed with traditional tools for nonlinear systems. One of our main contributions is that we establish stable formation and flocking on a group of agents with a second order integrator dynamics under control discontinuities.

The implemented path following controller is based on vector fields. This idea is not new; our contribution here is that we design this controller so that it does not need switching between different segments of the path and the controller is integrated seamlessly with formation control and flocking without the need for any leaders (actual or artificial).

Our simulations and experiments make use of a 2-D workspace with differentially steered mobile platforms but our framework can accommodate robots moving in 3-D spaces. Despite being developed as a centralized approach, it can be implemented in a decentralized fashion because if the communication graph is connected, the relative position information can be broadcasted to all the members, and the mathematical

Figure 1.1: Flocking made by a group of cranes. Photo courtesy of Shridhar Shah.

expression of the control law for a particular robot does not require the final destination of the others within the formation.

1.2.1 Flocking with Formation Control

Flocking is a behavior observed in some groups of social animals when they perform aggregate motion, usually over long distances. In this behavior, animals do not scatter in all directions, but rather, move as a single body, heading in the same direction while avoiding crashing into each other, see Fig. 1.1. Surprisingly, there is no evidence suggesting that the members of the flock communicate to synchronize their movements. Craig Reynolds [34] identified three basic component behaviors producing flocking when they are superimposed: *separation*, *alignment* and *cohesion*. Separation means individuals don't collide with each other, while alignment implies that the individuals move in the same direction. Cohesion simply means that the group is not going to break apart. Reynolds [34] implemented these behaviors as rules in computer simulations to replicate flocking.

We achieve flocking with a group of n robots located randomly over $\mathbb{R}^2$ while performing formation control and path following at the same time. In principle, the formation and path following controllers together can reproduce flocking behavior without the need for explicit velocity synchronization (alignment). Nonetheless, in this thesis we demonstrate that flocking with formation control can be also achieved independently, without necessarily having the flock to follow a designated path.

Formation control is in charge of the separation and cohesion stage, and path following takes care of the group's alignment along a route of interest. The formation controller employs an artificial potential field based on the relative positions between the agents, and because it is constructed in the framework of navigation functions, it has been shown [42] to guarantee convergence to the desired formation shape with collision avoidance. The formation controller permits almost arbitrary arrangements of relative positions between the robots so that the flock configuration reaches a specific desired shape asymptotically.

Early approaches to coordinated motion of several agents in the same workspace involved planning collision free routes for each individual robot. A limitation of such an approach is that it is not easily scalable with the size of the group. An alternative is to produce a single motion path describing the aggregate motion of all robots, then collision avoidance between robots becomes an issue. Another approach is to designate desired final locations for every robot in the group and then formulate an enlarged (simple,super-agent) motion planning problem [46]. However, for the formation planning problem, it is more important to achieve the desired relative distances among the robots than getting the formation located into a particular place in the workspace. In addition, once the formation at a given location, it is not clear how it is going to be moved as a single entity from that position.

Therefore, defining the problem in the relative position space and working with relative position vectors exclusively, may be more appropriate for the problem at hand. Relative positions also allows a mathematical modeling formulation that relates the robots and their relative positions to vertices and edges of a graph, respectively. This

model is a formation graph (see Definition 1) and encodes the formation specification we want for the group, too. The incidence matrix of the formation graph can be constructed by selecting the pairs of robots that are close to collide or based on the sensing capabilities of the available sensors. This matrix provides a mapping from positions with respect to an inertial frame to relative positions. The incidence matrix can serve as a instrumentation amplifier and provides a decoupling mechanism.

1.2.2 Formation Control Plus Path Following on the Fly

In a cooperative sensing task, such as inspection of an underwater pipeline or taking aerial photographs, falling into formation safely is not sufficient. To guarantee coverage of a specific area, the geometry of the most efficient path can be specified without requiring reaching a specific point at a specific time, design and implement a path following algorithm. Path following consists of designing controllers for a vehicle in order to follow a specified path as close as possible.

In general, implementing formation control and path following at the same time is difficult because the individual inputs for formation control and path following might contradict each other. In addition, the robots' nonholonomic constrains would not allow movement to certain directions. There is therefore a need for a method that can simultaneously accommodate both mission objectives.

Our path following sequence operates with coordinates relative to a particular fixed frame and the path segments are straight lines and Bézier line segments that work hand by hand with some special functions that makes the switching among the path segments unnecessary, see Chapter 6.

1.3 Thesis Organization

The overall architecture proposed consists of several components and subsequent chapters present these components in some detail.

First, we review the state of the art on each one of the main components of the proposed methodology in Chapter 2. Next, in Chapter 3 we present a formal description

for the problem we are solving and provide some key conceptual definitions for the mathematical analysis that follows. Chapter 4 contains the mathematical machinery to deal with nonsmooth differential equations and presents the existing results for converge analysis on systems with discontinuous right-hand sides.

Chapter 5 presents the methodology for formation control on systems with second order dynamics and Chapter 6 presents the integration of path following with formation control. The overall closed loop system is described in Chapter 7, which also includes the mathematical proof of convergence.

Finally, Chapter 8 describes how all the components of the control methodology physically interface with each other from the implementation point of view. Chapter 9 documents the simulations and experiments conducted to corroborate our approach, and Chapter 10 reflects our conclusions.

Chapter 2

LITERATURE REVIEW

The origins of a formal analysis of flocking behavior may be traced back to [34]. It consist of a computer animation model with multiple agents that implement three basic rules: Collision avoidance with nearby flockmates, velocity matching (attempt to match velocity with nearby members) and flock centering (attempt to stay close to neighbors). The purpose of this simulation model was to replicate the behavior observed in groups of animals like birds and fish without having to specify a trajectory for each individual in the group. Although flocking is observed in simulation, there is no formal mathematical proof that guarantees that such behavior will indeed emerge.

2.1 Formation Control

When a group of particles is required to move as a single entity, the particles need to keep constant relative positions; if it is necessary to make those particles adhere to a specific formation for the purposes of convenience (as for example, birds fly in V formation to exploit aerodynamic interactions), then these relative positions have to be specified. Refueling routines for planes or ships are another example of cases where a specific relative position need to be kept. But not all formation control approaches are compatible with flocking.

There are three dominant trends in approaching the formation control problem: Virtual structures, behavioral schemes, and potential field-based functions. In a virtual structure approach, a virtual vehicle is assigned to each real robot. The virtual formation (structure) expresses the ideal configurations of all vehicles at any given time instant. This structure allows for the formulation of a control law that makes all the virtual vehicles converge to a common value (Fig. 2.1). In a behavioral-based

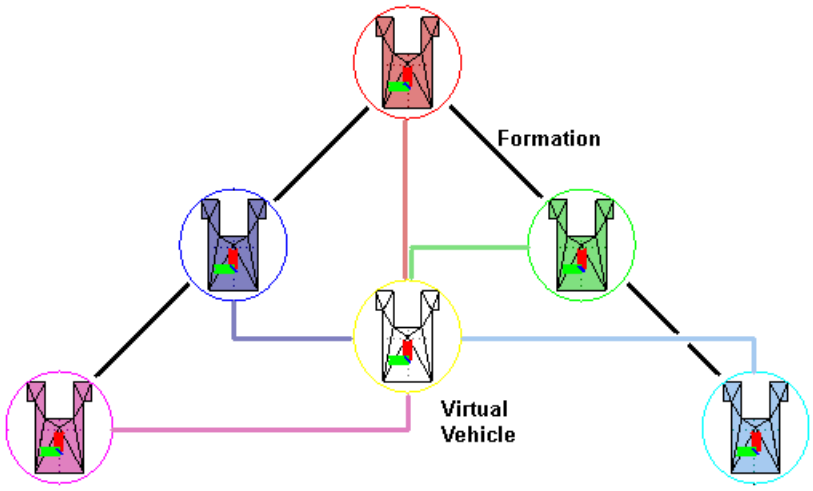


Figure 2.1: Virtual Vehicle-Structure approach. Each real robot is assigned a virtual vehicle and each virtual vehicle is controlled to stabilize on a common value.

approach, agents move based on a collection of rules like those formulated in the work of Reynolds (see [34]). The last main approach uses artificial potential fields to steer the robots into a specified formation that is expressed as the minimum of a potential function.

Balch and Arkin [1] explore a behavior-based approach to achieve robot formation-keeping. Agents are mechanically similar but distinguished from one another by a unique identification number, which is useful when setting each single robot in a formation and using leader-referenced formations. It is a behavioral approach with a set of rules for the robots like those proposed by Reynolds [34] and there is no formal proof for the composite behavior to satisfy any property. The authors report the implementation of two main reactive behaviors: *Motor Schemas* and *Steering* and *Speed Behaviors*; all the individual behaviors that compose them are executed as concurrent asynchronous processes.

Beard, Lawton and Hadaegh [2] propose a framework that incorporates the leader-following, behavioral, and virtual-structure approaches into a control schema for a multiple spacecraft interferometer. They use a block diagram to show the proposed

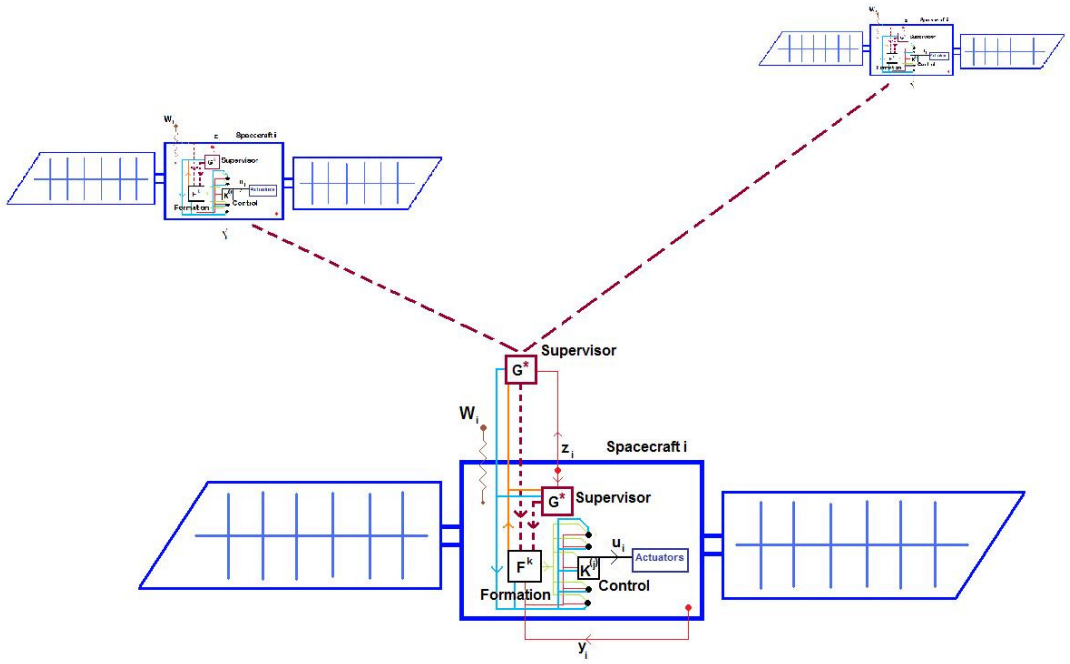


Figure 2.2: Architecture for spacecraft formation. Each spacecraft have all proposed behaviors implemented on the core and the supervisor module can be either centralized or decentralized.

relations among the local controllers of each spacecraft, the local formation control, the supervisor module and the spacecrafts themselves, (see Fig. 2.2). Note that there is feedback between all the module layers and that the supervisor module allows the complete approach to work as a centralized or decentralized fashion.

Each spacecraft module represents the spacecraft dynamics which has a force and torque control vector inputs with a disturbance vector input to account for external forces and torques; the output of the spacecraft dynamics are the position, attitude and a vector of performance variables used by the supervisor module. Next, the local controller of each spacecraft generates the force and torque control vector inputs for the spacecraft and brings together all the different control strategies needed when executing the different modes of operation dictated by the supervisor module. The local formation control module is the first coordination mechanism and it incorporates formation tasks such as initialization, translation, rotation and scaling of the formation pattern; it also

generates a formation performance vector used by the supervisor module. Finally, the supervisor module is in charge of selecting the appropriate formation control maneuver and the local controller for the spacecraft based on the outputs of the spacecraft and the formation performance vector.

Zavlanos and Kyriakopoulos [51] extend their previous work on centralized navigation, to decentralized multiple robot navigation. A particular robot R is driven from any initial configuration to a desired goal q_{Rd} in an environment where the other $n-1$ robots are considered to be mobile obstacles. Robot R occupies a set $R = \{q \in \mathbb{R}^2 : \|q - q_R\| \leq r_R\}$ where q_R and r_R are the position and radius of robot R, respectively. The ith obstacle (one of the other robots) occupies a disk $O_i = \{q \in \mathbb{R}^2 : \|q - q_{oi}\| \leq r_{oi}\}$ where q_{oi} and r_{oi} are the position and the radius of the obstacle. They propose the navigation function:

$$\varphi(q_R, q_o) = \frac{\gamma_d(q_R) + f(G(q_R, q_o))}{((\gamma_d(q_R) + f(G(q_R, q_o)))^k + G(q_R, q_o))^{\frac{1}{k}}}. \tag{2.1}$$

with control law:

$$\dot{q}_R = -K \frac{\partial \varphi(q_R, q_o)}{\partial q_R}$$

where q_o is a concatenated vector of all the $n-1$ mobile obstacles, $\gamma_d = \|q_R - q_{Rd}\|^2$ is a squared norm of robot R's position with respect to its destination, and $f(G)$ is a perturbation term that enables robot R to avoid a mobile obstacle passing through q_{Rd} even if robot R is already at q_{Rd}.

The authors employ a set of special metrics to formulate function $f(G)$. First, they use the term "relation" to describe the collision possibilities for a single agent, for instance, if the ith agent is colliding with one robot it will be referred as a relation "level-1" and a configuration of an agent colliding with three other robots will be known as a relation "level-3" and will be written as the set $(R_i)_3 = \{\{R_i, O_1\}, \{R_i, O_2\}, \{R_i, O_3\}\}$. The level of the relation expresses the number of robots involved in the collision . Next, a Relation Proximity Function is defined as:

$$(b_{R_j})_l = \sum_{m \in (R_j)_l} \beta_m$$

where $\beta_i(q_R) = \|q_R + q_{oi}\|^2 - (r_R + r_{oi})^2$ is a robot proximity function that vanishes at collision configurations. A Relation Verification Function is defined as:

$$(g_{R_j})_l = \begin{cases} (b_{R_j})_l + \frac{\lambda \cdot (b_{R_j})_l}{\left[(b_{R_j})_l + (B_{R_j^C})_l^{\frac{1}{h}}\right]}, & l \leq n-2 \\ (b_{R_j})_l, & l \leq n-1 \end{cases}$$

where λ, h are positive constants and $(B_{R_j^C}) = \prod_{m \in (R_j)_l^C} (b_m)_l$. Function G is given as:

$$G = \prod_{l=1}^{n_L} \prod_{j=1}^{n_{R,l}} (g_{R_j})_l.$$

Here n_L is the number of levels and $n_{R,l}$ is the number of relations in "level-l". Finally, function $f(G)$ is defined as:

$$f(G) = \begin{cases} a_0 + \sum_{j=1}^{3} a_j G^j & \text{, for:} 0 \leq G \leq X \\ 0 & , G > X \end{cases}$$

with constants:

$$a_3 = 2\frac{Y}{X^3} \quad a_2 = -3\frac{Y}{X^2} \quad a_1 = 0 \quad a_0 = Y$$

where X and Y are positive definite values of G and $f(G)$, respectively. They show in simulations that all the agents involved in the formation arrive to their destination. The robots already located in their desired configuration moved to allow another agent to pass. The approach is demonstrated in $\mathbb{R}^2$ and due to the robots being assumed disks, there can be at most seven in any collision configuration.

Olfati-Saber and Murray [30] work on a distributed control law for a group of robots with double integrator dynamics. They customize the definition of formation graph into the metric formation graph (MFG) as $G := (V_e, C, D)$ where $V_e = V \cup \{v_\infty\}$ is an extended set of the vertices V with an infinite vertex v_∞ in charge of casting out connections that are not between neighboring vertices, C is the connectivity matrix

and D is the distance matrix among the elements comprised on V_e with the same dimension of C. Additionally, they link the metric formation graph to a structural constrain function Φ given by:

$$\Phi(q) = \text{col}\{\phi_l\}_{l=1}^{s}$$
$$\phi(q) = \phi_l(q_i, q_j) = \|q_i - q_j\| - d_{ij}$$

where q_i is the position of the ith robot, d_{ij} is an element of the distance matrix D and there is a point $\bar{q} \in \mathbb{R}^{2n}$ called the equilibrium formation such that $\Phi(\bar{q}) = 0$. They propose the control law:

$$u(p, q) = -\nabla\langle\Phi(q), \Phi(q)\rangle - D(q, \hat{q})$$

where $D(q, \hat{q})$ is a damping force. They prove local collision-free asymptotic stability with unambiguous formation specification.

Ögren, Fiorelli and Leonard [26] design a distributed control law that configures the formation of a group of n robots according to a measured variable of interest (which is not specified) in an environment. This control law is based on M virtual bodies (vehicles) and artificial potentials. Their proposed control law u_i for a group of n robots with double integrator dynamics is

$$u_i = -\sum_{j=i}^{N} \nabla_{x_i} V_I(x_{ij}) - \sum_{l=1}^{M} \nabla_{x_i} V_h(h_{il}) - K\dot{x}_i$$
$$x_{ij} = x_i - x_j$$
$$h_{il} = x_i - b_l$$

where x_i are the coordinates of the ith robot, b_l are the coordinates of the lth virtual leader, V_I and V_h are potential functions that depends on the magnitude of the relative position between a pair of robots and the relative position between a robot and an a leader, respectively. In the expression of the control law, K is a positive definite matrix. A virtual structure specifies the desired translation, rotation and scaling of the formation, while an artificial potential is used to guarantee cohesion among the robots. They parametrize the virtual body motion with the scalar variable s to decouple

the formation stabilization stage from the maneuvering of the robots to do gradient climbing. After computing a least-squares approximation T_{est} of the gradient of the measured variable based on the information provided by the sensor of each robot have, they prescribe:

$$\frac{dr}{ds} = -\nabla(T_{est})$$

to make the virtual body moves in the direction of steepest ascent.

Ögren and Leonard [24] formulate a control law that drives a group of n robots into a formation while avoiding obstacles and moving through the workspace to arrive to a prescribed goal. They define a tree of the relative distances among the robots where the head of the tree is the leader, and propose a formation-leader obstacle set FLO as the obstacle set such that no vehicle will collide with a real obstacle if the leader stays out of FLO:

$$FLO(a,b) = \Theta\Big(\int\int_{\mathbb{R}^2} O(a+y_1, b+y_2)UR(y_1,y_2)dy_1dy_2\Big)$$

where $O \subset FLO \subset \mathbb{R}^2$ is the normal obstacle set, Θ is the Heaviside step function and UR is the region of uncertainty given by:

$$UR = \big\{r \in \mathbb{R}^2 | \exists i, \|r - r_{iref}\| \leq \sqrt{N}M(t)\big\},$$

Where r is the robot position vector, N is the number of edges of the tree, $r_{iref} = r_1 + d_{e1} + d_{e2} + \ldots + d_{eN}$ with $d_{ek} = d_{ij}$ (desired formation distances) and $M(t) = \beta(\|x(t_0)\|, t - t_0) + \gamma(\sup_{\tau \leq t} \|u(\tau)\|)$. As the expression for FLO is dependent heavily on the edges of the tree (the topology of the graph), the possibility of a switching tree topology is ignored.

Tanner and Kumar [45] present a navigation function that steers a group of mobile agents to achieve a particular formation (shape and orientations) while avoiding collisions with other robots and obstacles present in the environment. The authors consider the dynamics of the group of robots to be of the form $\dot{q}_i = u_i, q_i \in \mathbb{R}^n, i =$

$1, \ldots, N$ with a formation graph $G = (V, E, L)$ where $L = \{\gamma_{ij} | \gamma_{ij} = \|q_i - q_j - c_{ij}\|^2, c_{ij} \in \mathbb{R}^n, (v_i, v_j \in E)\}$ is a set of labels indexed by the edges of E. The proposed navigation function is given by:

$$\varphi(q) = \frac{\gamma_d(q)}{e^{\beta(q)^{1/\kappa}}} = \frac{\sum_{i=1}^{|E|} \gamma_{ij}(q_i, q_j)}{e^{(\prod_{i,k} \beta_{ik} \prod_{i,j} b_{ij})^{1/\kappa}}}, \quad i, j \in \{1, \ldots, N\}, k = 0, \ldots, s.$$

where the number of edges of the formation graph affect the terms of the navigation function. As collisions can occur between robots and between robots and obstacles, for all the possible combinations of an obstacle at p_t and agent $i = 1, \ldots, N$, an obstacle function is defined as:

$$\beta_{it} = \Big(1 - \lambda \frac{(\|q_i - p_t\|^2 - d^2)^2}{(\|q_i - p_t\|^2 - d^2)^2 + 1}\Big)^{\frac{sign(\|q_i - p_t\| - d) + 1}{2}}$$

Where d is a tunnig parameter so that β_{it} is smooth, $t = 1, \ldots, s$ and $\lambda = \frac{1+d^4}{d^4}$. The workspace boundary is also modeled as:

$$\beta_{i0} = \Big(1 - \lambda \frac{(R^2 - \|q_{ij}\|^2 - d^2)^2}{(R^2 - \|q_{ij}\|^2 - d^2) + 1}\Big)^{\frac{sign(\|q_i\| - R + d) + 1}{2}},$$

and b_{ij} as the obstacle function between a pair of robots $i, j \in \{1 \ldots N\}$:

$$b_{ij} = \Big(1 - \lambda \frac{(\|q_i - q_j\|^2 - d^2)^2}{(\|q_i - q_j\|^2 - d^2)^2 + 1}\Big)^{\frac{sign(\|q_i - q_j\| - d) + 1}{2}}$$

It is noted that it is not possible to ensure that the goal configuration is a non-degenerate critical point of the navigation function unless the labels of the formation graph edge are considered as configuration variables.

Dimarogonas, Kyriakopoulos and Theodorakatos [15] explore the formation control problem by enforcing communication restrictions and considering agents outside the sensing disk of the ith robot, which gives the system a hybrid nature. The analysis for stability is performed using Lyapunov stability techniques for hybrid systems. First they use I_i as the *information pattern* for agent i, which is defined as the set of agents in its sensing zone at each time instant:

$$I_i = \{j : \|q_i - q_j\| \leq d_C\}$$

In this way, the formation problem can be modeled as a switched system characterized by the *global information pattern* $I = \{I_1, \ldots, I_N\}$. Hence, both the navigation function and the control law can be written as:

$$\varphi_i(I_i) = \frac{\gamma_{di} + f_i}{((\gamma_{di} + f_i)^\kappa + G_i(I_i))^{\frac{1}{\kappa}}} \qquad u_i = -K_i \frac{\partial \varphi_i(I_i)}{\partial q_i}, i \in N$$

De Gennaro and Jadbabaie [12] study a group of robots with first order dynamics in a workspace with M obstacles and claim that when the topology of the interconnection among the robots is an acyclic graph, the minimum of the decentralized navigation function is unique. They propose to omit function $f(G)$ from their already proposed decentralized navigation function and settle for:

$$\varphi_i(q) = \frac{\gamma_i(q)}{(\gamma_i(q)^\kappa + \beta_i(q))^{\frac{1}{\kappa}}}$$

where γ_i is given by:

$$\gamma_i(q) = \sum_{j=1, j \neq i}^{N} \gamma_{ij}(q_i, q_j)$$

and γ_{ij} depends on then norm of the distance $\|r_{ij}\| = \|q_i - q_j\|$ and the desired final distance c_{ij}, in the form

$$\gamma_{ij} = \begin{cases} (\frac{z_{ij}}{c_{ij}})^2 & \|r_{ij}\| \leq c_{ij}; \\ \frac{a z_{ij}^2}{b z_{ij}^3 + c z_{ij}^2 + d z_{ij} + e} & c_{ij} < \|r_{ij}\| \leq R; \\ \frac{a(R - c_{ij})^2}{b(R - c_{ij})^3 + c(R - c_{ij})^2 + (R - c_{ij}) d + e} & \|r_{ij}\| > R. \end{cases}$$

The obstacle function is:

$$\beta_i(q) = \prod_{l \in C_i} \beta_{il}(q_i, q_l) \beta_{iw}(q_i)$$

where $\beta_{il}(q_i, q_l)$ is the obstacle function between agent i and agent of obstacle l and $\beta_{iw}(q_i)$ is the obstacle function between agent i and the boundary:

$$\beta_{il}(q_i, q_l) = \begin{cases} p_1(\|r_{il}\|) & \|r_{il}\| < \delta/4; \\ p_2(\|r_{il}\|) & \delta/4 \leq \|r_{il}\| < 3\delta/4; \\ p_3(\|r_{il}\|) & 3\delta/4 \leq \|r_{il}\| < \delta; \\ 1 & \|r_{il}\| \geq \delta \end{cases}$$

$$\beta_{iw}(q_i) = \begin{cases} 1 & \|q_{ij}\| \geq R_w - \delta; \\ p_I(\|q_i\|) & R_w - \delta < \|q_{ij}\| \leq R_w - 3\delta/4; \\ p_{II}(\|q_i\|) & R_w - 3\delta/4 < \|q_{ij}\| \leq R_w - \delta/4 \\ p_{III}(\|q_i\|) & R_w - \delta/4 < \|q_{ij}\| \leq R_w \\ 0 & \|q_{ij}\| \leq R_w \end{cases}$$

Here, p_i are third order polynomials in $\|r_{il}\|$ with unique minimum at collision. They prove that the proposed navigation function is non-degenerate at the desired formation and the critical points are in the interior of the workspace. Finally, there is a lower bound on the variable parameter κ such that the undesirable critical points are close to obstacles.

Loizou and Kyriakopoulos [20] work on the dynamics of a group of robots that have both holonomic and nonholonomic agents over $\mathbb{R}^2 \times (-\pi, \pi]$. After normalizing velocities, modeling the nonholonomic agents as unicycles, and concatenating the states of the holonomic and nonholonomic agents, the robot kinematics are:

$$\dot{p} = E \cdot u;$$

which is prolonged to

$$M\dot{\breve{u}} + R(p, \breve{u}) = f$$

where M and R represent the inertia matrix and Coriolis and centrifugal terms, E is given by:

$$E = \begin{bmatrix} V_h & 0 \\ 0 & C \end{bmatrix}$$

with $V = \text{diag}([\breve{u_h}^{\max^T} \breve{u_{nh}}^{\max^T}])$, $V_h \in \mathbb{R}^{3n \times 3n}$ is a submatrix of V containing the maximum velocities achievable by the holonomic subsystem and

$$C = \begin{bmatrix} C_{n+1} & 0 \\ \ddots & \\ 0 & C_m \end{bmatrix}; \qquad C_i = \begin{bmatrix} \breve{v_i}^{\max} \cos(\theta_i) & 0 \\ \breve{v_i}^{\max} \sin(\theta_i) & 0 \\ 0 & \breve{\omega_i}^{\max} \end{bmatrix}.$$

Instead of deriving a input-output linearization model for the nonholonomic robots, the authors propose to embed a dipolar like function into a multi-robot navigation function to implicitly control the orientation of the nonholonomic robots present in the system:

$$\varphi = \frac{\gamma_d}{(\gamma_d^\kappa + H_{nh} \cdot \beta)^{\frac{1}{\kappa}}}$$

Where H_{nh} has the form of a pseudo-obstacle

$$H_{nh} = \varepsilon_{nh} + \prod_{i=n+1}^{m} \eta_{nh_i}$$

and $\varepsilon_{nh} > \varepsilon_{\min} > 0$. Parameter η_{nh_i} is given by:

$$\eta_{nh_i} = ((q - q_d)^T \cdot n_{d_i})^2$$

where $n_{d_i} = [\underline{0}_{1\times 2(i-1)} \cos(\theta_{d_i}) \sin(\theta_{d_i}) \underline{0}_{1\times 2(m-i)}]^T$.

Finally, they arrive to the control law for the kinematic model of the form:

$$
\begin{aligned}
u_{xl} &= -\tau \cdot s(V_{xl}) \\
u_{yl} &= -\tau \cdot s(V_{yl}), \qquad l \in \{1, \ldots, n\} \\
\omega_l &= -\tau \cdot s(V_{\theta l}) \\
\omega_{x\mathrm{l}} &= -\tau \cdot s(\theta_{\mathrm{l}} - \theta_{nh\mathrm{l}}), \qquad \mathrm{l} \in \{M/N_\rho\} \\
\omega_{xl} &= -\tau \cdot s(V_{\theta j}), \qquad j \in N_\rho \\
v_i &= -\tau \cdot \frac{Z_i}{a_2 + Z_i} \cdot \mathrm{sgn}(V_{xi}\cos(\theta_i) + V_{yi}\sin(\theta_i)), \qquad i \in M \\
s(x) &\triangleq \frac{x}{a_1 + |x|}
\end{aligned}
$$

where V_x, V_y and V_θ are the derivatives of their Lyapunov candidate function along the x, y and θ directions. The authors take the previous control law, translate it from the kinematic model to a dynamic model and prove that the system is globally asymptotically stable.

Summers, Yu, Anderson and Dasgupta [39] investigate the formation of four robots with first order dynamics. Although the robot coordinates p are relative to a fixed frame, their control law is the negative gradient of a potential function $V(e(p))$ based on a defined error function of the interagent distances of the four robots:

$$
\begin{aligned}
r(p) &= \left[\|p_1 - p_2\|^2, \|p_1 - p_3\|^2, \|p_1 - p_4\|^2, \|p_2 - p_3\|^2, \|p_2 - p_4\|^2, \|p_3 - p_4\|^2\right]^T \\
e(p) &= r(p) - d^2 = [e_{12}, e_{13}, e_{14}, e_{23}, e_{24}, e_{34},]^T \\
V(e(p)) &= \frac{1}{2}\|e(p)\|^2
\end{aligned}
$$

It is claimed that there may exist equilibrium formation configuration with undesirable interagent distances when all the interagent distances are actively controlled, but any of those incorrect equilibrium configurations are locally unstable for formations with internal angles that satisfy an acuteness condition, i.e. when the edges and the diagonal segments of a goal formation make an acute angle. Hence, they conclude that the desired formation configuration is semi-globally asymptotically stable.

Zhang et al. [53] formulate a methodology that combines a behavior-based and leader-follower approach with artificial potential fields to navigate in an unknown environment. They consider situations in which the robots in the formation cannot continue moving because of some particular obstacle configuration, a situation which they call *locking*. Then, they introduce a *lock-prevent* strategy, and place the highest priority on it; the strategy is implemented by the followers. Each robot is embedded in a virtual shell, designed based on the robot's velocity and capability to brake. After input-output feedback linearization, a virtual leader is used and the focus is tuned in reducing the tracking errors between the virtual leader and the real followers. Their control law has the form

$$v_F = v_L \cos(\bar{\phi}) + k_1(\rho\cos(\varphi) - \rho^d \cos(\varphi^d + \bar{\phi})) + \rho^d \omega_L \sin(\varphi^d + \bar{\phi})$$
$$\omega_F = \frac{1}{L}\Big[v_L \sin(\bar{\phi}) - k_2(\rho\sin(\varphi) - \rho^d \sin(\varphi^d + \bar{\phi})) - \rho^d \omega_L \cos(\varphi^d + \bar{\phi})\Big]$$

where ρ^d and ϕ^d are the desired distance and the angle between a particular follower and the leader, v_L and ω_L are the linear and angular velocity of the leader. For the leader, the force of the target point is defined as:

$$F_{att} = -\nabla(k\|X - X_g\|^2) = -k(X - X_g)$$

with X and X_g as the position and desired configuration of the robot, respectively. The repulsion potential field due to an obstacle is:

$$U_{rep}(X) = \begin{cases} \eta\left(\frac{1}{\|X-X_g\|} - \frac{1}{\rho_0}\right)^2 & \|X - X_g\| \leq \rho_0 \\ 0 & \|X - X_g\| \geq \rho_0 \end{cases}.$$

where η is a positive ratio gain. ρ_0 is the biggest distance of obstacle affecting the robot.

Kwang-Kyo Oh and Hyo-Sung Ahn [27] focus on agents with limited motion capabilities. They assume that robots can move in only one of two ways: towards each other or by tracing a circle with respect to a static robot. The authors use only inter-agent distance information and coordinate their primitive movements in a sequential

manner. They prove that the target formation is achieved globally asymptotically stable if the information graph of the group is sequentially 3-valent (2-valent). Their formulation requires one of the robots to remain static while the others move, therefore this can be considered as a leader - follower approach.

Summers, Yu, Dasgupta and Anderson [40] work on minimally persistence formations, i.e. formations for which a directed graph $G = (V, E)$ with at least two vertices is minimally rigid: the graph has exactly $2|V| - 3$ edges and no vertex has more than two outgoing edges. They present a new result in center manifold theory:[1] They prove local exponential stability to the center manifold for a nonlinear dynamics of the formation $\dot{x} = f(x)$ with f being at least twice differentiable and a directed formation graph which makes the center manifold non-compact. Their control law restores a previously made formation when perturbed. Only one agent is responsible for maintaining each distance, and each control law is executed using only the relative position measurements of agents to which the agent must maintain its distance from.

Poonawala, Satici, Gans and Spong [33] present a time-invariant, state-feedback control law for two robots in which one is the leader and the other is the follower with a vision-based pose reconstruction system. Their contribution is due to the fact that their control law does not require any measurement or estimation of the leader robot velocity and leads to uniform ultimate boundedness of the error when the formation is moving in a general fashion. The controller also allows the variation of some of the constants to prioritize between the relative polar angle and the relative orientation. Their image-reconstruction method exploits restricted motions of the robots and the known robot's geometry and does not require singular value decomposition.

Coogan and Arcak [7] study the scaling of a formation for a group of robots with second order integrator dynamics with a leader-follower approach; all agents have knowledge of the formation specification but only the leaders have access to the scaling parameter λ, the followers have to estimate it. Two methodologies to achieve the

[1] A theory that deals with the stability of equilibrium points which after linearization gives one or more eigen-values with zero real part

formation are proposed: *single link method* and *multiple link method*. The control law for the single link method has the form

$$u_i = -\sum_{j=1}^{m} d_{ij}(z_j - z_j^d \lambda_i) - kv_i, \quad i \in I_F$$

where z_j is the relative displacement vector between two robots that are associated with edge j of the formation graph, v_i is the velocity of robot i, z_j^d is the desired formation specification, d_{ij} is an element of the incidence matrix, and k is a constant. For the single link method, λ_i is estimated as:

$$\lambda_i \triangleq \frac{1}{\|z_i^d\|^2}(z_i^d)^T z_i, \quad i \in I_F.$$

On the other hand, the control law for the multiple link method given by:

$$u_i = -\sum_{j=1}^{m} d_{ij}(z_j - z_j^d \lambda_j) - kv, \quad i \in I_F$$

with the estimated scaling parameter:

$$\lambda_j \triangleq \frac{1}{\|z_j^d\|^2}(z_j^d)^T z_j \quad j \in |E|$$

where E is the matrix of edges of the formation graph. The control law for the leaders in both cases is the same, it is given by:

$$u_i = -\sum_{j=1}^{m} d_{ij}(z_j - z_j^d \lambda) - kv_i, \quad i \in I_L.$$

Note that on the multiple link method the indices of the estimated λ_j are associated to the links of the measurement graph instead of the agent as in the single link method. Finally, the authors prove stability of their closed loop system by using the small gain theorem.

Kwon and Chwa [19] formulate a hierarchical formation control composed by line formation control and column formation control. The subgroups of robots generate line formations that later make up the overall formation structure using the column

formation control. This hierachical control adopts a leader-follower strategy between robots and between subgroups of line structures;[2] furthermore, it employs a vector field to make the followers track the leaders, i.e. the same vector field formulation is used for the follower robots to track the leader robot and for the follower line structures to track the leader structure. One of the disavantages is that the authors assign real robots as leaders, so if a leader fails, the formation is compromised. Another disavantage is that there is no formation feedback among the objects (robots, line structures) that compose this approach.

All these approaches solve crucial aspects of the formation control problem, making different assumptions about the capabilities of the agents, the range of the sensors and the information structure used to embed the formation specification. However, there are topics that are not covered, and which we need to have addressed to realize the goals of this thesis; for instance, contributions such as [2] and [26] do not indicate if they leave room for non-degenerate equilibrium formations. Approaches like [2], [30], [24], [45] neglect the switching nature involved on either the transition of behaviors or the sensors' range. Contributions such as [2], [15], [51], [30], [14], [45], [15], [12], [39], [27] and [7] assume either a first or second order integrator dynamics and neglect the nonholonomic constraint present in common wheeled robots: robots arrive to the formation with orientations that might not necessarily point towards a possible desired common direction. As a result, robots would break up the formation when trying to align to a common velocitity vector. Finally, some contributions introduce elaborate constructions of the collision avoidance controller, leading to inconvenient expressions[3] of the final controller [51], [45], [12], [20].

---

[2] There is a local robot leader per each line structure and there is a line structure leader which is followed by the other line structures.

[3] mathematical expressions that take excessive processor time because they consider collision among all $n(n-1)/2$ pairs of objects

2.2 Flocking

Flocking is understood as the behavior in which all the members of a group move with the same vector velocity (which implies no collisions among the entities) and constant relative positions asymptotically. We regard flocking as the core behavior for a mobile sensor network and we want to merge formation control and path following to it. Hence, it is important to review the state of the art to build on, rather than replace existing solutions.

Tanner, Jadbabaie and Pappas on [44] and [43] generate stable flocking motion for a group of robots for both fixed and dynamic topologies in a two paper. First, they analyze stability using a fixed neighboring graph topology and point out that the only way in which a fixed topology is able to guarantee collision avoidance is when the agents are interconnected to each other, so the neighboring graph is complete. For the second part, they used a neighboring graph which varies in time giving rise to a dynamic control interconnection topology and a switching control law. The authors employ a nonsmooth version of LaSalle's principle proposed by [38] to show convergence. Flocking motion is still guaranteed provided that the connectivity in the neighboring graph is maintained.

As the information flow in a flock is directed and the graph topollogy of the agents involved goes through changes, Olfati-Saber and Murray [29] address consensus problems for networks of robots with switched and fixed topologies. The authors present two consensus protocols,[4] provide a convergence analysis and test on directed networks with fixed and switching topologies, and undirected networks with communication time-delays and fixed topology. The authors demostrate that there is a tradeoff between the robustness to time delays and the performance of achieving a consensus in a network: the maximum time-delay tolerated by a network under a certain linear consensus protocol is inversely porportional to the maximum degree of nodes of the network.

[4] A consensus protocol is a control law that influences the value of the vertices to agree, i.e. if x_i denotes the value of a node v_i in a network, then nodes v_i and v_j agree if $x_i = x_j$.

Cucker and Smale [11] explore different scenarios and conditions to achieve flocking in a population of birds that are moving in $\mathbb{R}^3$. They extend a discrete time model from [10] to a continuous model in which every bird adjusts its velocity by adding a weighted average of the differences of its velocity with those of other birds.

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -L_x v, \end{aligned} \tag{2.2}$$

where x_i and v_i are the position and velocity of the ith bird, and L_x is the Laplacian of the adjancency matrix. The adjacency matrix is given by:

$$a_{ij} = \frac{H}{(1 + \|x_i - x_j\|^2)^\beta},$$

Here, H and β are fixed constants. Then, the authors state that an unique solution of dynamics (2.2) with initial conditions x_0 and v_0 will tend to a common limit $\hat{v}$ and a limit vector $\hat{x}_{ij}$ asymptotically, if $\beta < 1/2$. To guarantee the same solution of (2.2) with $\beta \geq 1/2$, specific conditions over x, v, and the Fiedler eigenvalue[5] of L_x should be imposed, see [11, Theorem 2].

The authors also extend their analysis on situations in which some pairs of agents do not communicate and situations where there is connectivity among the agents but the connection channels changes (a dynamic topology).

Tanner, Jadbabaie and Pappas [41] provide the formal definition for flocking used on this document. They employ an undirected velocity graph $G_c = \{V, E_c\}$ with E_c representing communication links and a directed position graph $G_s = \{V, E_s\}$ with E_s representing sensing links. Moreover, they motivate the use of nonsmooth analysis because of the changing nature of the robots' network. They consider second order integrator dynamics for which the input is the addition of alignment and cohesion components. They prove flocking under the assumption that the switching present in the robot network can occur arbitrarily fast and the graphs are always connected.

[5] The second eigenvalue.

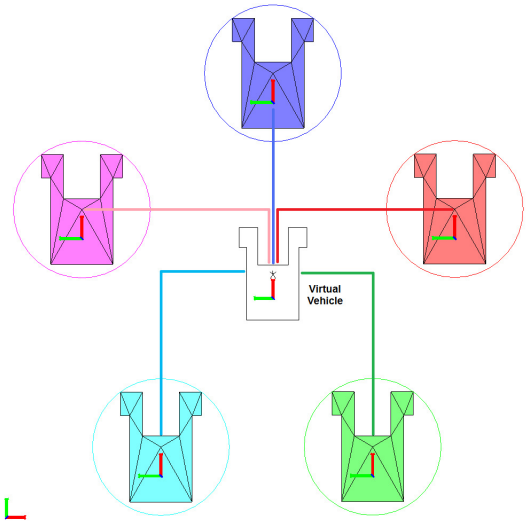


Figure 2.3: Formation with a Virtual Vehicle (In the center). Notice the relative position of each robot with respect to its virtual vehicle as lines of color; it is a different relation specification for each of them.

Yoshioka and Namerikawa [50] propose several control laws such as consensus, flocking, and flocking with collision avoidance, based on virtual structures. Their information graph (similar to the one defined in Definition 1) is a continuous strongly connected graph, and switching is not considered. The authors link the formation problem to a consensus problem by pointing out that a virtual vehicle (with a particular specification) is assigned to each real robot, and this virtual vehicle has to converge to a common position and orientation (Fig . 2.3). Flocking is identified as being more challenging as a problem by noting that while consensus requires relative positions, flocking requires both relative positions and relative velocities between the virtual vehicles.

Zavlanos, Tanner, Jadbabaie and Pappas [52] propose on a controller that guarantees flocking and at the same time maintains network connectivity. They consider the switching nature of a undirected qualitative proximity graph $G(t) = (V, E(t))$ so

the proposed control law is:

$$u_i(t) \triangleq -\sum_{j\in N_i(t)} (v_i(t) - v_j(t)) - \sum_{j\in N_i(t)} \nabla_{x_i}\varphi_{ij}(t)$$

$$\varphi_{ij} \triangleq \begin{cases} \|x_{ij}\|_2^{-2} + P_1(x_{ij}), & \|x_{ij}\|_2 \in (0, r] \\ 0, & \|x_{ij}\|_2 \in (r, R-r) \\ \frac{1}{R^2 - \|x_{ij}\|_2^2} + P_2(x_{ij}), & \|x_{ij}\|_2 \in [R-r, R) \end{cases}$$

where φ_{ij} is a symmetric artificial potential function in charge of collision avoidance and edge preservation, x_{ij} is the relative position between two robots i and j, v_i is the velocity of robot i, $P_1(x_{ij})$ and $P_2(x_{ij})$ are second order polynomials in $\|x_{ij}\|$. The authors state that their control law experiences discontinuity and induces switching nonlinear dynamics. To ensure flocking while preserving connectivity, they prove there is a subset of the workspace which is invariant and solutions of the differential equations cannot escape, which implies that the network topology is connected for those times, regardless of the discontinuity previously mentioned. Their proximity graph is related to the communication and it is assumed that robots can sense each other is positions regardless of the distance. A distributed topology algorithm that deletes and creates links among the agents based on a defined partition of the spatial neighborhood of an agent (Fig . 2.4). The authors point out that the creation is not as demanding as the deletion of links, agents share information about their local connectivity to develop estimates of the global topology and formulate an *auction-based* framework to agree what links to delete. The deletion process is done by deleting links one by one always making sure of the connectivity of the graph.

Martin, Fazeli, Jadbabaie and Girard [21] study flocking behavior conditions by adopting different approach. Instead of formulating a control law and prove either stability or convergence, they consider the radius that agents i and j are able to communicate at time t to be a random variable and focus their analysis on setting a bound on the probability for the agents to achieve flocking. They state that the bound should depend on the initial position and velocities of the robots and consider a discrete time dynamics in which the communication among robots is modeled with

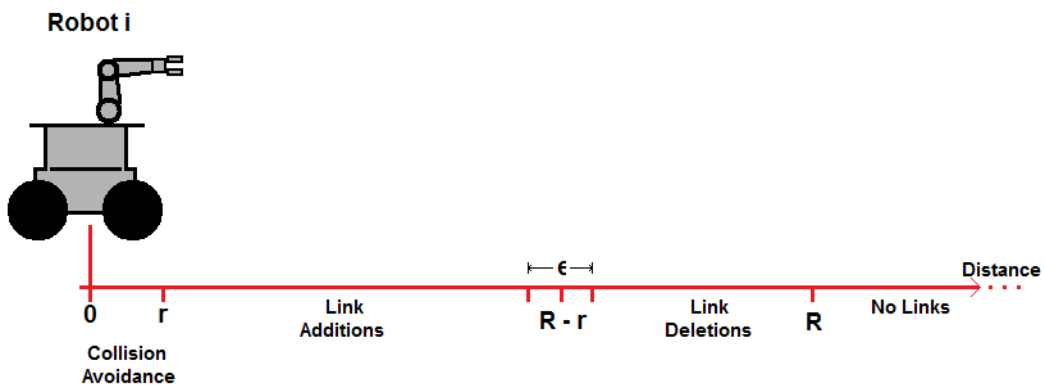


Figure 2.4: Partition of the neighboring region of the *ith* robot. Note that each φ_{ij} operates according to this partition.

a graph that does not have cycles (a tree). The edges of the graph are encoded in the elements of a time-varying adjacency matrix

$$a_{ij}(t) = 1_{\|x_i(t)-x_j(t)\| \leq R_{ij}(t)}$$

The authors point out that the second smallest eigen-value of the Lapplacian matrix is a good measure of the connectivity of the graph. Their main result states that given $\rho > 0, \tilde{\lambda}_2 > 0$ and $\|\delta(0)\| \leq \frac{\rho\tilde{\lambda}_2}{\sqrt{2}}$, the probability that flocking occurs is given by:

$$1 - \frac{1}{1+\kappa^2} \text{ where } \kappa = \frac{1}{\nu_\varepsilon}\Big(\frac{\rho\tilde{\lambda}_2}{\sqrt{2}\|\delta(0)\|} - 1\Big)^2$$
$$\nu_\varepsilon = \frac{\varepsilon(\mathbb{E}(\lambda_2(\tilde{L}(t)))^2 - \tilde{\lambda}_2^2)}{2\tilde{\lambda}_2 - \varepsilon\mathbb{E}(\lambda_2(\tilde{L}(t)))^2} \in [0,1]$$

where ρ is an estimate of the robustness of the bound on the relative distance of two agents and δ is a velocity disagreement vector.

Di Paola, De Asmundis, Gasparri and Rizzo [32] formulate a decentralized architecture for preserving connectivity in a multi-robot system with robots equipped with hetereogeneous sensors and having limited field of view. That enforces the use of a directed graph. Usually, most of the works focus on undirected graphs from which

a minimum spanning tree is derived as the structure to represent the problem. However, when working with directed graphs, the authors claim it is necessarily to work with an (approximation of a) *minimum strongly connected spanning subgraph*, which is the equivalent to a minimum spanning tree for directed graphs. For their analysis they establish a field of view $\Phi_i(t)$ for each robots based on the different sensing capabilities. The authors assume a limited wireless communication radius, equal for all robots. They also consider a dynamic communication graph $G(t) = \{V, \tilde{E(t)}\}$ which is undirected and represent the communication links among the robots and a dynamic sensing graph $D(t) = \{V, E(t)\}$ which is directed and $E(t) \subseteq \tilde{E(t)}$. The set $\Phi_i(t)$ is given by:

$$\Phi_i(t) = \{p \in \mathbb{R}^2 : \|p - p_i(t)\| \leq \rho_i, |\theta(t) - \theta_i(t)|\}$$

involving only relative positions and the angular position of each robots is measured with respect to a fixed reference axis. As the communication radius can be bigger or equal to the sensing radius, the deletion and addition of links is based on graph $D(t)$. The key idea is to delete a direct link between two robots if there is a longer alternative path connecting those two robots, so it will look for cycles on the dynamic sensing graph and delete links in such a way that there is strong connectivity along the graph. They divide their methodology in four stages: Cycle Identification, Out-neighborhood Evaluation, In-neighborhood Request Evaluation and Agent Update, which are executed in a synchronous way. Ther resulting framework is essentially a behavior-based approach; there is not a formal proof showing the desired features.

Flocking approaches like [1] and [32] present a set of rules that when applied, they seem to achieve flocking, but without any formal proof that the behavior will indeed emerge; References such as [11] and [50] do not consider switching. Approaches such as [52] perform the analysis with either first or second order dynamics, but the possible nonholonomic nature of the robots is not taken into account.

2.3 Path Following

Path following is an integral part of our control framework because our group of robots moves, once assembled, along a desired path in order to search an area of interest.

Sarkar, Yun and Kumar [37] distinguish trajectory tracking from path following by pointing out that a set of points are given in trajectory tracking controllers are designed to make the system to reach those points at the required times; in path following, on the other hand, the complete geometry of the path is specified and it is more important to follow the path closely than hitting specific waypoints at specific times.

Sørdalen and Canudas de Wit [28] propose a piecewise smooth feedback control law for path following based on their previous work (see [5]). In that previous work [5], they define a family of circles that lie on a local y axis and are tangent to the origin, the circles are intended to drive the robot to the origin in a circular path by means of a change of variables a and α given by:

$$a(x, y) = \frac{x^2 + y^2}{y} \arctan(y/x)$$

$$\alpha(x, y, \theta) = (\theta - \theta_d)(1 - 2\pi n).$$

where a is the arc length from the robot position to the local origin, α is an orientation error, and x, y and θ are the robot's position and its orientation. Then, a local reference frame proposed by [5] is used over each starting point p_i of each path segment to be traversed.

Nelson, Barber and McLain [22] develop a method for path following for miniature air vehicles through the construction of vector fields surrounding the path. They develop vector fields for primitive geometrical entities such as straight lines, circular arcs and orbits and use Lyapunov stability arguments to demonstrate asymptotic stability. They combine the previous geometrical shapes to generate complex paths and point out that one element at a time should be enabled to avoid the chance of having multiple sinks, singularities and dead zones.

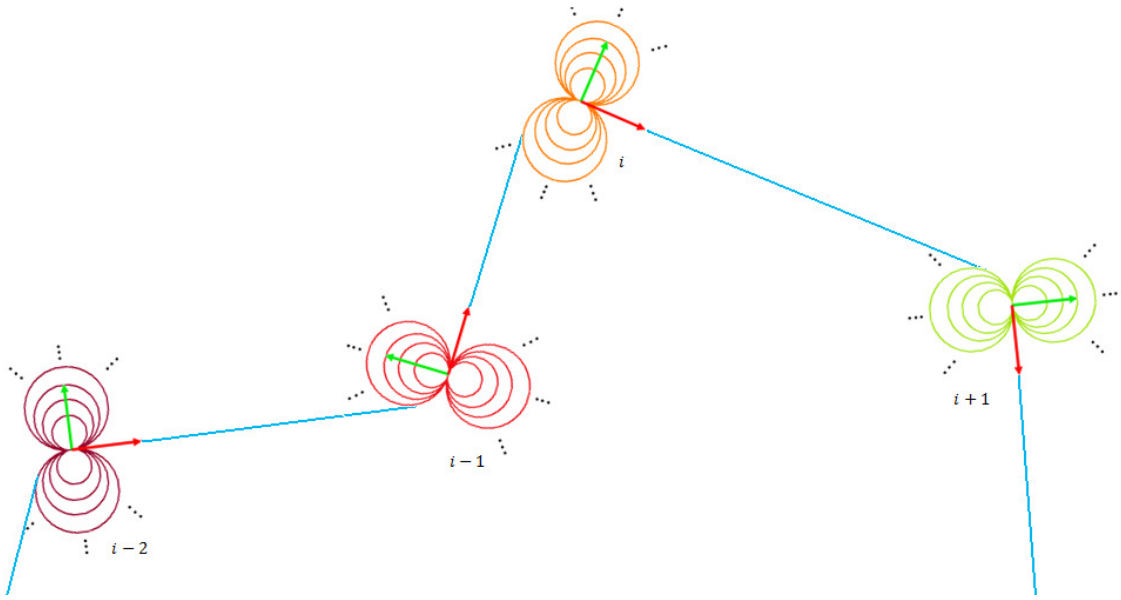


Figure 2.5: Sørdalen and Canudas de Wit path following (cf. [28, Fig 2.]).

It can be observed that even when the final vector field that is used to induce path following is smooth, approaches [28] and [22] require a switching mechanism to enable only a single path segment to be evaluated at a time, in order to avoid sinks, dead zones and singularities.

2.4 Conclusions

Reviewing the state of the art, it is observed that a prevailing assumption is that of smoothness either on their sensing and communication graphs or during the change of ruling behavior. However, in implementations of multiagent coordination algorithms, we frequently observe discontinuous and nonsmooth behavior.

In addition, several robot platforms are nonholonomic, while the majority of work on flocking assumes either first order or second order integrator dynamics and there are no approaches that can seemlessly combine flocking with formation control and path following.

Despite the fact that formation control with virtual structures is a dominant

approach, integration of virtual structures to path following routines is not always desirable because it requires a different control input for each robot when trying to minimize the formation error when the virtual structure has to traverse through a curvilinear path. As the goal point per each robot in the virtual structure is continuously moving, the approach requires the consideration of the overall dynamics in the form of a time-varying system.

In this thesis, the goal is to find a global control law that steers a group of n robots into a prescribed set of relative positions and having their relative velocities converge to zero while the group of robots follows a designed path as a single entity.

As path following, formation control and flocking are all components of the desired solution, we propose the decomposition of the overall problem into subproblems to make the analysis more amenable. With this in mind, we need:

- An formation control algorithm that steers the group of robot into a prescribed set of constant relative position and guarantees collision avoidance among the robots.
- A path following control algorithm that drives a particular common point of the formation attained by the previous algorithm to follow a designed route.
- A decoupling mechanism that avoids the interference among the control laws of each subproblem and keeps time invariance.
- The velocity vector of each robot to converge to a common value or equivalently, the relative velocity among the robots has to converge to zero.

This subproblems are addressed in Chapter 5, Chapter 6 and Chapter 7 repectively.

Chapter 3

PROBLEM STATEMENT

3.1 Objectives

The goal is to formulate a framework in which a group of robots, ramdonly located over the Euclidean space ${}_sE$ fall into a specified formation without colliding, and then continue moving together as a rigid body following a prescribed path while maintaining the formation they made before. For this type of control problems, it is generally required to define a formation (sensing) graph; we use the definition of a directed formation graph $G = \{V, E, C\}$ given by [42] as:

Definition 1 (Formation Graph)**.** *$G = \{V, E, C\}$ is a directed formation graph consisted of:*

- *A set of vertices $V = \{v_1 \dots v_n\}$, indexed by the mobile agents.*
- *a set of edges $E = \{(i, j) : \{1 \dots n\} \times \{1 \dots n\}\}$ containing ordered pairs of nodes that represent interagent position specification, and*
- *a set of labels $C = \{c_{ij} | (i, j) \in E\}$.*

The formation graph $G = \{V, E, C\}$ embeds the state of the formation (relative positions of the robots) in the set E and keeps the formation specification on set C. Set C can be expressed as a stack vector of the desired relative position vectors (a quantity with magnitude and direction); hence, the orientation of the formation is already considered. Now, for the problem statement, we study the case of differential robots in the plane. Nonetheless, our methodology can be extended to mobile robots with more degrees of freedom.

Problem Statement 1 (Flocking with formation control and path following)**.** *For a group of n robots deployed over an Eclidean space E, represented by the vector state*

$$p = [x_1 \quad y_1 \quad \theta_1 \quad \dots \quad x_i \quad y_i \quad \theta_i \quad \dots \quad x_n \quad y_n \quad \theta_n \quad]^T \tag{3.1}$$

where x_i, y_i and θ_i are the coordinates of the ith robot (3.3) and $p \in \mathbb{R}^{3n}$. We need a global control law that makes the robots come together into a formation specification embedded in a formation graph given by Definition 1 where the set C is of the form

$$c = [q_{1x} \quad q_{1y} \quad q_{2x} \quad q_{2y} \quad \dots \quad q_{(n-1)x} \quad q_{(n-1)y}]$$

and keep the formation while the group moves as a rigid body along a prescibed path composed by straight and curved line segments.

3.2 Assumptions

One of our assumptions is the nonslip condition. Take as an example two bodies F and M in relative motion that are in contact at point I (see Fig 3.1), then there exist a normal vector $\hat{n}_{12}$ to a plane that is tangent to both bodies at point I. Then, body M has five degrees of freedom to move with respect to body F and the velocity screw of body M seen from body F at point I is given by:

$$\left\{\mathcal{V}_{\frac{M}{F}}\right\}_I = \left\{\begin{matrix} \omega \\ v_{I\in M/F} \end{matrix}\right\}$$

where the term $v_{I\in M/F}$ is the velocity of point I "instantaneously" attached to body M seen from body F, and represents the slip velocity of body M seen from body F. If the movement of M with respect to F is without slip for a finite time, then

$$v_{I\in M/F} = 0$$

which is known as the nonslip condition. Taking into account the nonslip condition specific to wheeled robots (Fig. 3.2) and assuming rigid wheels on our robots, the velocity screws of the wheels at the points of contact c_1 and c_2 are

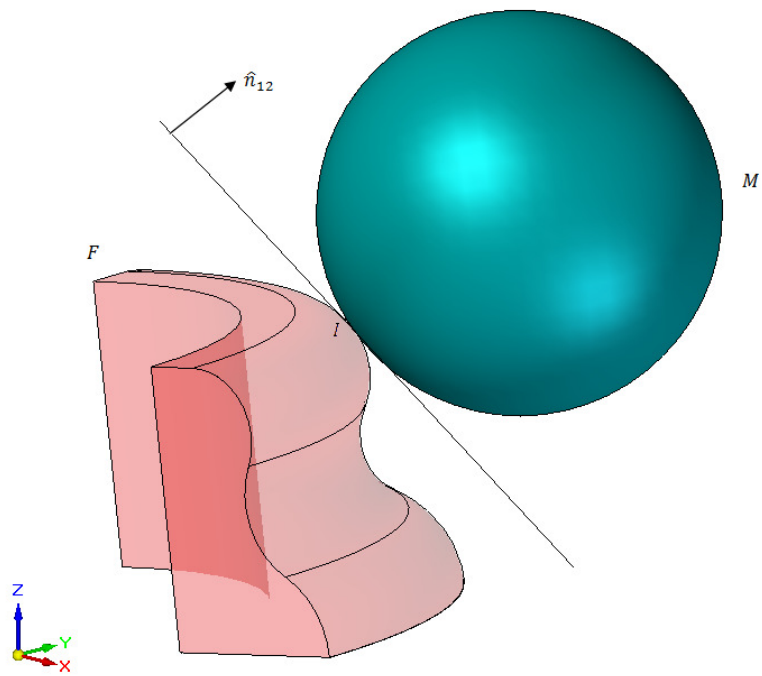


Figure 3.1: Bodies F and M in relative motion at contact point I.

$$\left\{\mathcal{V}_{\frac{wl}{E}}\right\}_{c_1} = \left\{\mathcal{V}_{\frac{R}{E}}\right\}_{c_1} + \left\{\mathcal{V}_{\frac{wl}{R}}\right\}_{c_1} = \left\{\begin{smallmatrix}\Omega_1\\0\end{smallmatrix}\right\}_{c_1}$$
$$\left\{\mathcal{V}_{\frac{wr}{E}}\right\}_{c_2} = \left\{\mathcal{V}_{\frac{R}{E}}\right\}_{c_2} + \left\{\mathcal{V}_{\frac{wr}{R}}\right\}_{c_2} = \left\{\begin{smallmatrix}\Omega_2\\0\end{smallmatrix}\right\}_{c_2}$$

where Ω_1, Ω_2 are the total angular velocities of wl and wr, and wl and wr stand for left wheel and right wheel respectively. The velocity screw of the robot seen from frame E is:

$$\left\{\mathcal{V}_{\frac{R}{E}}\right\} = \left\{\begin{matrix}\omega\hat{z}_E\\ \dot{x}\hat{x}_E+\dot{y}\hat{y}_E\end{matrix}\right\}_o = \left\{\begin{matrix}\omega\hat{z}_E\\ \dot{x}\hat{x}_E+\dot{y}\hat{y}_E-\frac{\omega d}{2}\hat{x}_E\end{matrix}\right\}_{c_1} = \left\{\begin{matrix}\omega\hat{z}_E\\ \dot{x}\hat{x}_E+\dot{y}\hat{y}_E+\frac{\omega d}{2}\hat{x}_E\end{matrix}\right\}_{c_2}$$

and the velocity of the left wheel and right wheel with respect to the robot are:

$$\left\{\mathcal{V}_{\frac{wl}{R}}\right\}_{o_1} = \left\{\begin{matrix}\omega_l \hat{y}_R \\ 0\end{matrix}\right\}_{o_1} = \left\{\begin{matrix}\omega_l \hat{y}_R \\ -r\omega_l \hat{x}_R\end{matrix}\right\}_{c_1}$$

$$\left\{\mathcal{V}_{\frac{wr}{R}}\right\}_{o_2} = \left\{\begin{matrix}\omega_r \hat{y}_R \\ 0\end{matrix}\right\}_{o_2} = \left\{\begin{matrix}\omega_r \hat{y}_R \\ -r\omega_r \hat{x}_R\end{matrix}\right\}_{c_2}$$

As the relative velocity on the point c_1 and c_2 instantaneously attached to wl and wr seen from E must be zero, we have

$$\begin{aligned} v &= \sqrt{\dot{x}^2 + \dot{y}^2} \\ v - \frac{wd}{2} - r\omega_l &= 0 \\ v + \frac{wd}{2} - r\omega_r &= 0 \\ v &= \frac{r}{2}(\omega_l + \omega_r) \\ \omega &= \frac{r}{d}(\omega_r - \omega_l). \end{aligned}$$

The equations above relate the forward and angular velocity of the robots to the angular velocities of the wheels. The nonslip condition also implies that there is no velocity along the axis perpendicular to the axis which forms an angle θ with the initial frame, that is:

$$-\dot{x}\sin(\theta) + \dot{y}\cos(\theta) = 0. \tag{3.2}$$

The constraint (3.2) is a nonholonomic constraint.

3.3 Overview of the Approach

The first step taken toward the development of our approach is to express Problem 1 in a more convenient way. Consider the motion in an inertial frame E (O,$\hat{x}_E$,$\hat{y}_E$),

of one robot (o, $\hat{x}_r$, $\hat{y}_r$) (Fig. 3.2). Based on the analysis of Section 3.2, the kinematics of that robot in the point o is given by:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{3.3}$$

where $\dot{x}$ and $\dot{y}$ are the velocities in the $\hat{x}_E$ axis and $\hat{y}_E$ axis respectively, θ is the angular position of the robot with respect to $\hat{x}_E$ and v and ω are the linear and angular velocity input respectively. We apply a prolongation: we consider inputs v and ω to be states and consider their derivatives to be the new inputs:[1]

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \\ \dot{v} &= \Gamma \\ \dot{\omega} &= \Omega \end{aligned} \tag{3.4}$$

Based on [37], we select a point h on the mobile robot that is along the axis $\hat{x}_r$, displaced a distance L from the origin of the frame fixed to the robot. The coordinates of h with respect to the frame E are:

$$\begin{aligned} x_h &= x + L\cos\theta \\ y_h &= y + L\sin\theta, \end{aligned}$$

We chose the output for this system to be $y = [x_h \quad y_h]^T$, and carry out partial feedback linearization for the states x, y and θ:

$$\begin{aligned} z_1 &= x_h \\ z_2 &= y_h \end{aligned}, \qquad \dot{z} = \nabla(y) \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

[1] Prolongation: make the inputs states and introduce updates for these extra states.

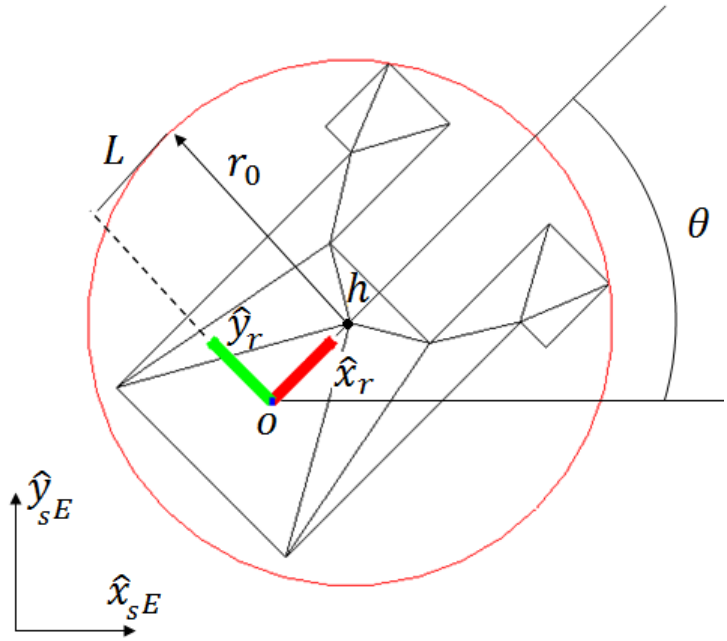


Figure 3.2: Model to deduce the kinematics of a differential robot. Note point h along the $\hat{x}_r$ axis.

$$\begin{aligned} \dot{z}_1 &= \cos\theta v - L\sin\theta\omega \\ \dot{z}_2 &= \sin\theta v + L\cos\theta\omega \\ \dot{v} &= \Gamma \\ \dot{\omega} &= \Omega \end{aligned} \quad \rightarrow \quad \begin{aligned} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} cos\theta & -L\sin\theta \\ sin\theta & L\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \\ \dot{v} &= \Gamma \\ \dot{\omega} &= \Omega \end{aligned}$$

with the linearized inputs:

$$\begin{aligned} v &= v_{navx}\cos\theta + v_{navy}\sin\theta \\ \omega &= \frac{1}{L}\big(v_{navy}\cos\theta - v_{navx}\sin\theta\big). \end{aligned}$$

The output dynamics are:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v_{navx} \\ v_{navy} \\ \Gamma \\ \Omega \end{bmatrix} \tag{3.5}$$

Note that the orientation is implicitly controlled through the inputs calculated from the group path following. This input-output feedback linearization allows to concatenate all the robots' states and describe them in the form of second order integrator dynamics. Consider the ith robot with linearized states z_{1i}, z_{2i}, v_i, ω_i and inputs Γ_i and Ω_i we have:

$$x_{1i} = \begin{bmatrix} \dot{v}_i \\ \dot{\omega}_i \end{bmatrix} \qquad x_{2i} = \begin{bmatrix} z_{1i} \\ z_{2i} \end{bmatrix}. \tag{3.6}$$

Then, our problem statement becomes:

Problem Statement 2 (Flocking with formation control and path following)**.** *For a group of n robots deployed over an Euclidean space E, represented by the transformed vector state*

$$x = [x_{11} \quad x_{12} \quad \dots \quad x_{1i} \quad \dots \quad x_{1n} \quad x_{21} \quad x_{22} \quad \dots \quad x_{2i} \quad \dots \quad x_{2n}]^T \tag{3.7}$$

where x_{1i} and x_{2i} are the linearized coordinates of the ith robot (3.6). We seek a global control law that make the robots fall into a formation specification embeded in a formation graph given by Definition 1 where the set C is of the form

$$c = [q_{1x} \quad q_{1y} \quad q_{2x} \quad q_{2y} \quad \dots \quad q_{(n-1)x} \quad q_{(n-1)y}]$$

without colliding and keep the formation while the group moves as a unified body towards a prescibed path composed by straight and curved line segments.

3.3.1 Final Remarks

We have expressed the original representation of our group of robots (3.1) into a linearized version (3.7) that is more convient due to the fact it is expressed as a second order dynamics and the orientation of the robots is implicit.

The formation control problem requires a group of n robots with dynamics (3.5) to move towards an equilibrium configuration prescibed by the formation graph $G = \{V, E, C\}$ previously defined. To solve this problem, we build on the methodology

developed by [42], which consists of a kinematic formation controller based on navigation functions (see Chapter 7) and we extend it to double integrators dynamics and include a flocking control input component.

We add another potential field-based component to generate a path following algorithm compatible with formation control. This path field methodology is constructed by adding a set of geometrical primitives that contain a vector field which is evaluated based on a group of customized windowing functions (see Chapter 6). The reason this path following methodology works is because it applies a uniform control input to all robots and thus does not affect the internal dynamics configuration of the formation.

The key element that permits the formation control with flocking and the path following control to work in a orchestrated fashion is the incidence matrix of the complete graph. The contruction of this matrix provides the decoupling mechanism that allows the execution of both methodologies without interference (see 7.3).

Chapter 4

MATHEMATICAL PRELIMINARIES

Several discontinuous, nonsmooth phenomena can manifest themselves during the application of a multiagent control scheme. This is because of the range of the positioning sensors, range and function of the communication hardware, and the theoretical methodology chosen to tackle the problem. Here, we focus on the robots which are the closest to a collision configuration and make use of the minimum distacne function $d(\cdot) = \min(\cdot)$ to evaluate an artificial potential field proposed in [42]. Discontinuty in the potential field comes from the min function. Therefore it becomes necessary that we make use of tools form discontinuous nonsmooth analysis.

In the determination of minimum distances, we make use of graph theoretic concepts. Specifically, the elements considered in the argument of $d(\cdot)$ are selected from the weighted edges of the complete super-graph of the formation graph. For this reason, we review in this chapter some concepts to study graphs.

4.1 Nonsmooth Analysis

Nonsmooth phenomena occur naturally. For instance, take the case of cable networks [17]; a cable network is an assembly of a finite number of cables which can only transmit tensile forces, but not compression forces. Figure 4.1 of [17] shows the nonsmooth character of this response. Other examples include a static object going from the static friction coefficient to the dynamic one, the velocity of a bouncing ball, cost functions which are composed by a min of max function, electric circuits composed by diodes, transistors and threshold phenomena.

Now, consider the vector differential equation

$$\dot{x} = f(x, t) \tag{4.1}$$

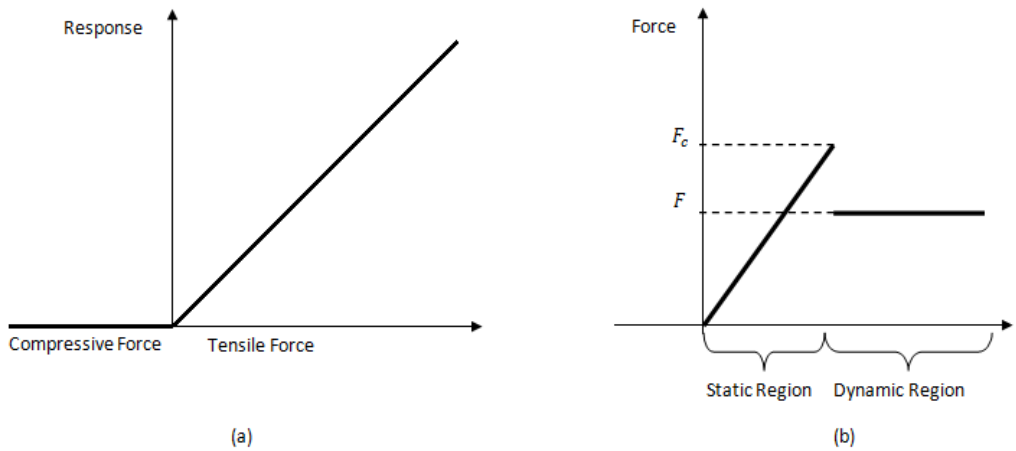


Figure 4.1: Examples of nonsmooth behavior. (a) Response of a member on a cable network. (b) Friction coefficient example.

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$. When the right hand side of a differential equation (4.1) describing the evolution of a physical phenomenon is discontinuous, the dynamics of the process is described with a differential inclusion:

$$\begin{aligned} \dot{x}(t) &\in F(t, x(t)) \\ F(t, x(t)) &= \text{co}\{\lim f(x_i) | x_i \to x, x_i \notin N_f \cup N\} \end{aligned} \tag{4.2}$$

where co is the convex hull (see Section 4.1.2), N and N_f are sets such that for all $N \subset \mathbb{R}^m, \mu N = 0$. The operator μ is the Lebesgue measure[1] and N_f is a set in which $f(x, t)$ is not defined; see [38].

According to Cortes [9], a Filippov solution of (4.1) on $[0, t_1] \subset \mathbb{R}$ is an absolutely continuous map $x : [0, t_1] \to \mathbb{R}^n$ that satisfies (4.2) for almost all $t \in [0, t_1]$.

Let Y be a subset of X. A function $f : Y \to R$ satisfies the Lipschitz condition on Y if, for some nonegative scalar K:

$$|f(y) - f(y')| \leq K\|y - y'\| \tag{4.3}$$

[1] It is a standard way to assign a measure to a subset contained in a n-dimensional Euclidean space; if $n = i$, with $i = 1, 2, 3$, the Lebesgue measure correspond to length, area and volume respectively.

for all points in Y. When we say that f is Lipschitz near x, we mean that f is locally Lipschitz at x.

4.1.1 Generalized Directional Derivative

The following definition is from [6]:

Definition 2. *Let f be Lipschitz near a given point x in $X \subset \mathbb{R}^n$, let y be any other point near x, v an arbitrary vector and t is a positive scalar. The Generalized Directional Derivative of f at x in the direction of v, denoted by $f^\circ(x, v)$ is defined as:*

$$f^\circ(x, v) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{f(y + tv) - f(y)}{t}. \tag{4.4}$$

This function has several interesting properties:
Let f be Lipschitz near x with constant K. Then

- The function $v \to f^\circ(x, v)$ is finite, positive homogeneous, and subadditive on X, and satisfies $|f^\circ(x, v)| \leq K\|v\|$.
- $f^\circ(x, v)$ is upper semicontinous[2] as a functions of (x, v) and, as a function of v alone, is Lipschitz with constant K on X.
- $f^\circ(x, -v) = (-f)^\circ(x, v)$.

The generalized directional derivative allows us to calculate the rate of change of an nonsmooth function along a specific direction.

4.1.2 Generalized Gradient

The generalized gradient is the equivalent of the regular gradient for the case where functions are not differentiable. The generalized gradient of f at x, denoted $\partial f(x)$, is the subset of X^* (the dual space of X) given by

$$\big\{\zeta \in X^* : f^\circ(x, v) \geq \langle \zeta, \xi \rangle \text{ for all } v \text{ in } X\big\}.$$

[2] A real valued function $f(x)$ at point x_0 is upper semicontinous at x_0 if for every $\epsilon > 0$, there exist a neighbourhood U of x_0 such that $f(x) \leq f(x_0) + \epsilon$ for all x in U.

with $\|\zeta\|_*$ as the norm defined in X^* as:

$$\|\zeta\|_* := \sup\{\langle \zeta, v\rangle : v \in X, \|v\| \leq 1\}$$

and B_* as an open unit ball in X^*.

In finite-dimensional spaces, we can also express the generalized gradient as:

$$\partial f(x) = \mathrm{co}\Big\{ \lim_{i\to\infty} \nabla f(x_i) : x_i \notin S \cup \Omega_f \Big\},$$

defined over a ball εB located at x. Ω_f is a set of points in $x + \varepsilon B$ for which f is not differentiable and S is any set of measure zero. The generalized gradient $\partial f(x)$ is the convex hull of all points of the form $\lim \nabla(f_i)$, where $\{x_i\}$ is any sequence which converges to x while avoiding $S \cup \Omega_f$.

While the gradient is a vector, the generalized gradient is a convex set. A set C is convex if the line segment between any two points in C lies in C, it means that for any c_1 and $c_2 \in C$ and any $0 \leq \theta \leq 1$, we have $\theta c_1 + (1-\theta)c_2 \in C$, see [3]. Now, the convex hull co of a set C is the set of all convex combinations of points in C:

$$\mathrm{conv}(C) = \Big\{ \theta_1 c_1 + \ldots + \theta_k c_k \mid c_i \in C, \theta_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \theta_i = 1 \Big\}.$$

For example, the magenta area on Fig. 4.2 is the convex hull of set $C = \{\vec{c}_1, \vec{c}_2, \vec{c}_3\}$ where $\vec{c}_1 = [2, 0, 0]^T$, $\vec{c}_2 = [0, 1, 0]^T$ and $\vec{c}_3 = [0, 0, 1]^T$. In the figure, $\vec{c}_1$ is the red vector, $\vec{c}_2$ is the green vector and $\vec{c}_3$ is the blue vector.

Let f be Lipschitz of rank K near x. Then

- $\partial f(x)$ is a nonempty, convex, weak*-compact subset of X^* and $\|\zeta\|_* \leq K$ for every ζ in $\partial f(x)$.
- For every v in X, one has

$$f^\circ(x, v) = \max\{\langle \zeta, v\rangle : \zeta \in \partial f(x)\}.$$

The last property reveals how the generalized directional derivative and the generalized gradient are related to each other.

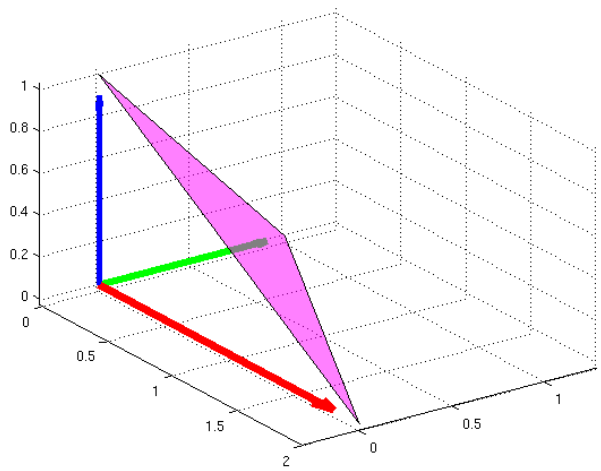


Figure 4.2: Convex Hull of set $C = \{\vec{c}_1, \vec{c}_2, \vec{c}_3\}$. The magenta area represents the region of the combination $\alpha_1\vec{c}_1 + \alpha_2\vec{c}_2 + \alpha_3\vec{c}_3$ such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

4.1.3 Basic Calculus on Generalized Gradients

For the statements below, we still assume that all functions are locally Lipschitz.

Proposition 1 (Scalar Multiples)**.** *For any scalar s, one has*

$$\partial(sf)(x) = s\partial f(x).$$

Proposition 2 (Local Extrema)**.** *If f attains a local minimum or maximum at x, then $0 \in \partial f(x)$.*

Proposition 3 (Finite Sums)**.**

$$\partial\Big(\sum f_i\Big)(x) \subset \sum \partial f_i(x).$$

Corollary 1. *Equality holds in Proposition 3 if all but at most one of the functions f_i are strictly differentiable at x.*

Corollary 2. *For any scalars s_i, one has*

$$\partial\Big(\sum_{i=1}^{n} s_i f_i\Big)(x) \subset \sum_{i=1}^{n} s_i \partial f_i(x)$$

and equality holds if all but at most one of the f_i are strictly differentiable at x.

The following is the definition of a Bannach space, from [13].

Definition 3. *Let $(E, \|\cdot\|)$ be a normed space and let $\{x_n\}$ be a sequence of points of E. We shall say $\{x_n\}$is a Cauchy sequence if the limit, as m and n tend to infinity, of $\|x_n - x_m\|$ is zero. We say that $(E, \|\cdot\|)$ is a complete normed space, or that $(E, \|\cdot\|)$ is a Banach space, if every Cauchy sequence of points of E converges to a point of E.*

Corollary 3. *Any finite dimensional normed space is a Banach space*

Definition 4. *f is said to be regular at x provided*

- *For all v, the usual one-sided directional derivative $f'(x; v)$ exists.*
- *For all v, $f'(x; v) = f^\circ(x; v)$*

Chain Rule I

One has

$$\partial f(x) \subset \mathrm{co}\Big\{\sum \alpha_i \zeta_i : \zeta_i \in \partial h_i(x), \alpha \in \partial g(h(x))\Big\}$$

and equality holds under any one of the following additional hypotheses:

- g is regular at $h(x)$, each h_i is regular at x, and every element α of $\partial g(h(x))$ has nonnegative components. (In this case the $\overline{co}$ is superfluous.)
- g is strictly differentiable at $h(x)$ and $n = 1$. (In this case the $\overline{\mathrm{co}}$ is superfluous.)
- g is regular at $h(x)$ and h is strictly differentiable at x. (In this case it follows that f is regular at x, and the $\overline{\mathrm{co}}$ is superfluous.)

Chain Rule II

Let F be a map from X to another Banach space Y, and let g be a real-valued function on Y. Suppose that F is strictly differentiable at x and that g is Lipschitz near $F(x)$. Then $f = g \circ F$ is Lipschitz near x, and one has

$$\partial f(x) \subset \partial g(F(x)) \circ D_s F(x).$$

Equality holds if g (or $-g$) is regular at $F(x)$, in which case f (or $-f$) is also regular at x. Equality also holds if F maps every neighborhood of x to

4.1.4 Ryan's Invariance Principle

Ryan's invariance theorem allows us to conclude the weak convergence of the solutions of a differential inclusion to a set, see [36].

Consider the non-autonomous initial-value problem

$$\dot{x}(t) \in F(t, x(t)), \quad x(t) \in G, \quad x(t_0) = x^0, \tag{4.5}$$

where $G \subset \mathbb{R}^N$. We need the following definitions:

Definition 5. *A solution x of the differential inclusion (4.5) is said to be maximal, if it does not have a proper right extension which is also a solution of (4.5).*

Definition 6. *Every solution of (4.5) can be extended to a maximal solution.*

Definition 7. *A solution $x \in AC([t_0, \omega; G])$ of (4.5) is precompact if it is maximal and the closure $cl(x([t_0, \omega)))$ of its trajectory is a compact subset of G.*

Definition 8. *From [23]. Let $x(t)$ be a bounded solution of (4.5). If x_t defined by $x_t(0) = x(t + 0)$ has no positive limit points on the boundary of X, then $x(t)$ is precompact.*

Theorem 1 (Ryan's Invariance Theorem c.f. [36])**.** *Let $V : G \to \mathbb{R}$ be locallly Lipschitz. Define*

$$u : G \to \mathbb{R}, \quad z \to u(z) := \max\{V^\circ(z, \phi) | \phi \in X(z)\}$$

Suppose that $U \subset G$ is non-empty and that $u(z) \leq 0$ for all $z \in U$. If x is a precompact solution of (4.5) with trajectory in U, then, for some constant $c \in V(cl(U) \cap G)$, x approaches the largest weakly-invariant set in $\Sigma \cap V^{-1}(c)$, where

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

4.2 Graph Theory Preliminaries

A graph G is a mathematical object defined as a pair (V, E). The set V is called the vertex set of G and its elements are called vertices. The set E is known as the edge

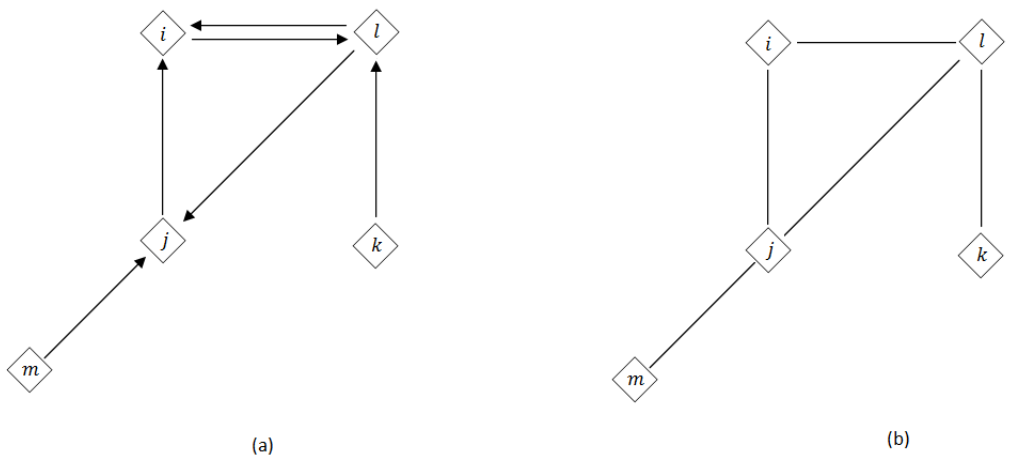


Figure 4.3: (a) A directed Graph. (b) An undirected Graph

set of G, its elements are called edges and define connections among the vertices of G. Labelled graphs can have additional components, i.e., functions that map vertices of edges of the graph to symbols or numbers. Tanner, Jadbabaie and Pappas [44] point out that the only way in which a fixed topology is able to guarantee collision avoidance is when the agents are all interconnected to each other; De Gennaro and Jadbabaie [12] claim that when the topology of the interconnection among the robots is an acyclic graph, the minimum of the decentralized navigation function is unique.

4.2.1 Directed and Undirected Graphs

A directed graph, also known as a digraph (see [8]), is a pair (V, E) where V is a finite set and E is a binary relation on V that is ordered, i.e. there can be an edge going from node i to j but not from j to i, see Fig. 4.3. Self-loops among a directed graph are possible. On the other hand, an undirected graph $G = (V, E)$ is composed by unordered edges so having an edge connecting i to j is equivalent to have a connection from j to i, see Fig. 4.3.

Graphs have been used to model problems such as cost evaluation for the deployment of a communication network over a region, optimization routes on integrated

systems of transportation, routing path of pcb's (Printed Circuit Boards). So we can use an analogy for graphs. A path of lenght k between two nodes i, j that belong to V in a graph is a sequence of vertices $\langle v_0, \ldots, v_k \rangle$ such that $i = v_0$ and $j = v_k$. A path $\langle v_0, \ldots, v_k \rangle$ forms a cycle if $v_0 = v_k$ and contains at least one edge.

A graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \in V$ and $E' \in E$.

4.2.2 Representation of Graphs

There are two ways two represent a graph $G = (V, E)$: using an adjacency matrix or an adjacency list. For the adjancency-matrix representation of a graph we make use of a matrix $A = (a_{ij})$ of size $|V| \times |V|$ where $|V|$ is the cardinality of set V (the number of vertices on G) and a_{ij} is equal to one if the edge $(i, j) \in E$ or zero otherwise. A different representation for directed graphs known as the $|E| \times |V|$ incidence matrix consists of marking a node with 1 if the node is the head of the edge, -1 if the node is the tail of the edge and 0 otherwise.

For the adjancency-list representation of a graph we use an adjacency list per any i vertex in V which contains all vertices that are neighbors of vertex i. In the case of a directed graph, the adjacency list gives account of all the vertices that are in the head of all the edges which have vertex i as tail. An undirected graph is connected if every pair of vertices is connected by a path. A directed graph is strongly connected if every two vertices are reachable form each other.

Sometimes, the edges of the graph not only represent the presence of a link between two nodes but also contain values that weights the links. Then, that type of graph is known as a weighted graph.

Chapter 5

FORMATION CONTROLLER DESIGN

We extend the first order integrator dynamics of Boddu and Tanner [42] to a second order integrator dynamics. In Chapter 7 we demostrate the convergence properties of the combined scheme, given the already available kinematic formation control scheme. Therefore, the purpose of this chapter is to review that kinematic formation controller.

We choose to use an artificial potential field because it unifies three basic separate stages of robot navigation, see [35]: Path planning, that is, the formulation of a free obstacle curve connecting the goal with the initial position of the robot based on geometric data; Trajectory planning, that is the generation of a particular parametrization based on the curve obtained on the previous step; and finally, robot control, which is the formulation of a controller to make the physical robot follow the reference trajectory as closely as possible. We particularly chose to use the artificial potential field proposed by [42].

For a group of N robots involved in the formation, the structure of the artificial potential field in [42] is:

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

where $\gamma(q)$ is a positive function that encodes the distance of the system from its desired configuration and $\beta(q)$ is a scalar, positive simi-definite function which is zero only at collision configurations. The workspace is given by

$$\mathcal{Q} = \mathbb{R}^{nN(N-1)/2} - \{q : \|q\| < d_0, \forall (i,j) \in N\}$$

taking into account that the maximun number of possible edges in the formation graph defined in Definition 1 is $\frac{N(N-1)}{2}$ and n is the dimension in which the robots are moving (2-D of 3-D). Workspace $\mathcal{Q}$ is decomposed into:

- $\mathcal{Q}_0 \triangleq \{q \in \mathcal{Q} | 0 < \beta(q) < \varepsilon\}$, a region close to a collision configuration.
- $\partial\mathcal{Q}$ is the boundary of the free space.
- $\mathcal{Q}_2 \triangleq \mathcal{Q} \setminus (\partial\mathcal{Q} \cup \mathcal{Q}_0)$.

Note that (5.1) does not fit into the formal definition of a navigation function because it is not analytic and the range of the function is not $[0, 1]$.

5.1 A Non-smooth Potential Function with Global Convergence Properties

The proposed $\gamma(q)$ in [42] is given as

$$\gamma(q) = \gamma_d^{\kappa}(q), \text{ with } \gamma_d(q) \triangleq \sum_{(i,j)\in E} \|q - c\|^2$$

where κ is a tunning parameter and c is the prescribed formation embbeded on the formation graph, see Definition 1.

The proposed $\beta(q)$ is:

$$\beta(d) = \log\left(\mu - ae^{-(-r+d+d^2)^2}\right), \text{ with } d(q) = \min(\|q_i\|)$$

where r and a are tunning parameters. The proposed navigation function is

$$\varphi(q) = \frac{\gamma(q)}{\beta(q)} = \frac{\left(\sum_{(i,j)\in E} \|q - c\|^2\right)^{\kappa}}{\log\left(\mu - ae^{-(-r+d+d^2)^2}\right)} \tag{5.2}$$

The term $\min(\|q_i\|)$ picks out the most critical distance (the smallest one) among the robots, avoiding the multiplication of pair-wise agent collision metric functions, and resulting in a switching formation graph which introduces non-smooth behavior to φ.

As an example, consider the movement of three robots in a plane, see Fig. 5.1. The relative position between robots 1, 2 and 3 are represented by q_{12}, q_{13} and q_{23}. In Fig. 5.1 (a), the lines in red, blue, and green represent the relative position q_{12}, q_{13} and

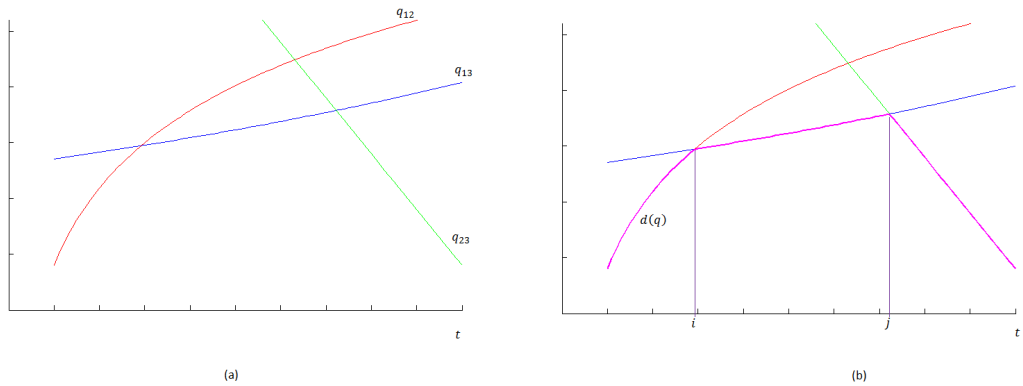


Figure 5.1: Nonsmooth feature embedded on φ due to $d(q)$.(a) Graph of q_{12}, q_{13} and q_{23} . (b) $d(q) = \min(\|q_i\|)$, note the nonsmoothness at points i and j.

q_{23} over the time, respectively. Now, in Fig. 5.1(b), as min() picks the minimum value (norm) out of q_{12}, q_{13} and q_{23}, it generates a nonsmooth line which is represented by the purple line; note that this happens on points i and j. Before point i, the min() function takes the values of q_{12}; but after point i, min() takes the values of q_{13}. The nonsmoothness arises because q_{12} and q_{13} are evolving at a different rate of change from one another and min() changes its value with the slope of the value that is the minimum.

5.2 Properties of the Artificial Potential Field

Among other properties for function φ, Boddu and Tanner [42] demostrated that:

- The destination point, the formation specification c is a nondegenerate minimum of $\bar{\varphi}$.
- If the formation specification is valid, all critical points of φ are in the interior of $\mathcal{Q}_2$ and arbitrarily close to $\mathcal{Q}_0$ by tunning k.
- Even when φ is not a navigation function, it still has some of its properties, φ is polar and it has the structure of a navigation function:

$$\bar{\varphi}(q) = \varphi(q)^{\frac{1}{\kappa}}$$

5.3 Gradient-Induced Control Law

Boddu and Tanner [42] proved that for the case of $\dot{q} = u$, the controller $u = -\mathrm{Ln}(\partial(\varphi))$ (see below a definition for $\mathrm{Ln}(\cdot)$ function) guarantees almost global asymptotic stability for the desired formation configuration. In Chapter 7 we will analyze the stability properties of the following controller, assuming the structural properties of ϕ guaranteed by the construction method of [42].

Let $q = x_2$ and $\dot{q} = x_1$. We propose the following control law for the case when φ is differentiable at x_2:

$$u = -\nabla(\varphi(x_2)) - x_1$$

The negative gradient is in charge of steering the robots to the prescribed formation configuration and drive the robots away from each other when approaching a collision configuration. Finally, $-x_1$ forces velocity synchronization among the agents and provides damping.

When φ is not differentiable at x_2, the control law becomes:

$$u = -\mathrm{Ln}(\partial\varphi)(x_2) - x_1$$

where $\mathrm{Ln}(\cdot)$ is a set valued map that assings to each subset of the generalized gradient $\partial\varphi(x_2)$ the set of least-norm elements in the closure of $\partial\varphi(x_2)$; for a convex and closed set S, Ln(S) maps to a singleton which is the orthogonal projection of the zero vector on S.

Chapter 6

FORMATION PATH FOLLOWING

The approach of deploying an artificial potential field around geometrical primitives such as straight lines and curved segments is not new. Some approaches of this type require switching among the elements that build up the path to avoid kinematic singularities. However, our construction makes use of windowing functions (see [31, Chapter 7]) to formulate a path following algorithm which does not require switching (Fig. 6.1).

The windowing functions will enable us to smoothly add up path segments (Fig. 6.2). Figures 6.2 (a) and (b) show a blue pulse signal and a red triangular signal with their respective windowing function in magenta and green, the windowing functions are over the regions of the functions we are interested in keeping when merging the signals. We multiply each signal with its respective windowing function and we add them up after. Figure 6.2(c) shows the results of the merged signal, where the smooth transition from the signal of part Fig. 6.2(a) to that of Fig. 6.2(b) can be observed.

Through the use of appropiate windowing functions, we are able to produce a single, smooth vector field capable of steering the system (6.2) along the entire reference path. The input used given to the system for path following consists of two components, which are superimpossed. The first component, u_s is the contribution due to the n straight line path segments present in the entire reference path and the second component, u_b is the contribution of the m Bézier line path segments present in the total reference path:

$$u_p = u_s + u_b = \sum_{i=1}^{n} u_{s,i} + \sum_{j=1}^{m} u_{b,j} \tag{6.1}$$

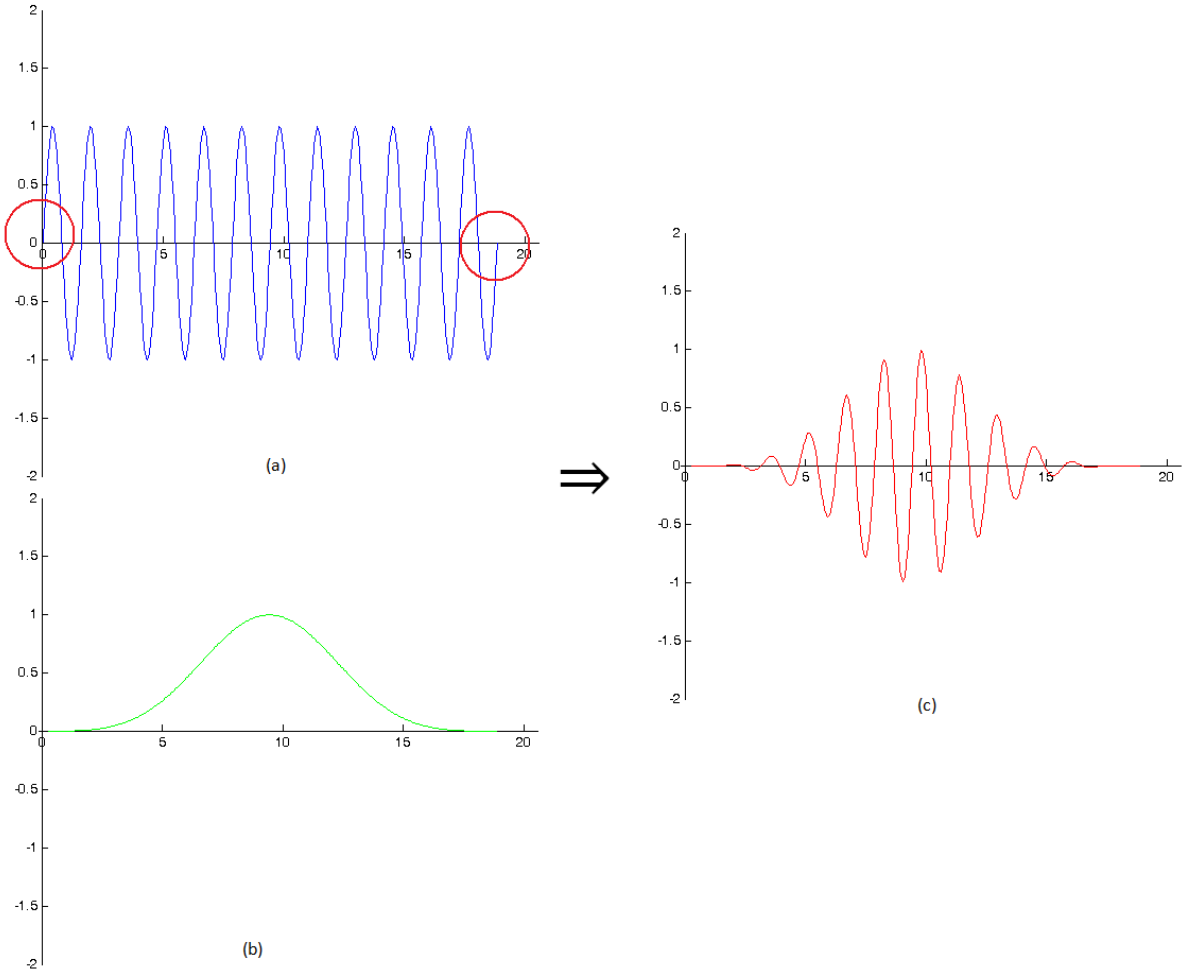


Figure 6.1: Original use of windowing functions. The purpose of windowing functions in signal processing is to mask the discontinuity in frequency both at the beginning and end of the recorded signal, see the red circles in (a). A windowing function displayed in (b) multiplies the original signal to obtain the processed signal, in (c). Note that the windowing function tends to zero at the end points of the signal attenuating the efect of the discontinuity.

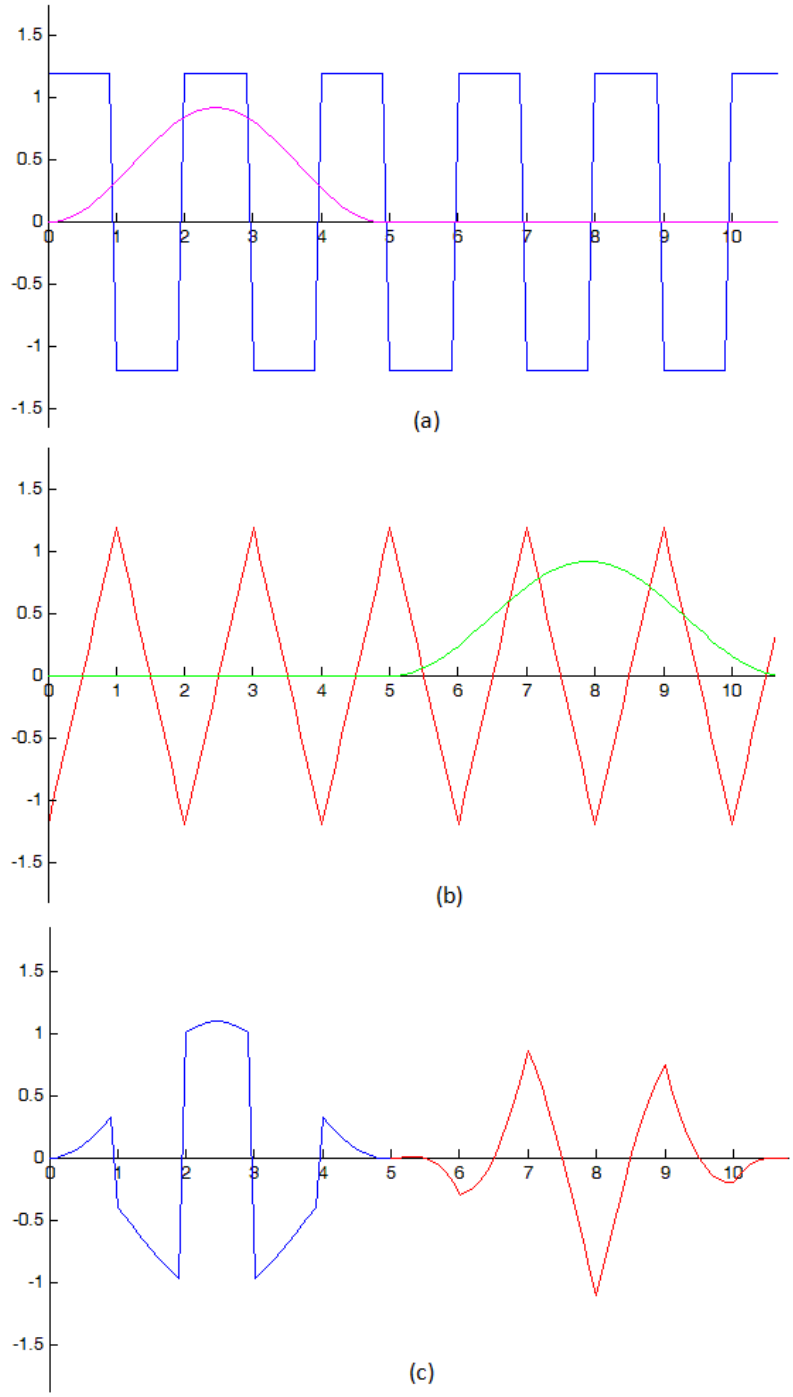


Figure 6.2: Use of windowing functions to merge signals without introducing discontinuities.

The problem considered here is to steer the geometric center of the formation, denoted as x_{cm} along the designated path. We use a proportional controller and take advantage of the internal damping effect of the $-x_1$ term present in the formation controller. The path following control input is applied uniformly to all vehicles, and because it is a common bias to all vehicles, it has no effect on the relative position dynamics. We considered dynamics are:

$$\dot{x}_{cm} = u_p \tag{6.2}$$

where x_{cm} is the coordinates of the formation's geometric center of mass, which is steered using u_p as a common control input reference applied uniformly to all members of the group.

6.1 Following straight line segments

Consider the configuration of Fig. 6.3, which shows a straight line path going from point P_0 to P_1 in $\mathbb{R}^3$ where $p = (x, y, z)$ is the location of the formation's geometric center, $\vec{v} = P_1 - P_0$, $\hat{v}$ is the unit vector of $\vec{v}$. We also have:

$$\begin{aligned}
\overrightarrow{P} &= p - P_0 \\
\overrightarrow{m} &= \hat{v} \times (\hat{v} \times \overrightarrow{P}) \\
\overrightarrow{n} &= \overrightarrow{m} \times \hat{v} \\
q_s &= \|\overrightarrow{P}\| \sin\left(\arccos\left(\frac{\overrightarrow{P} \cdot \hat{v}}{\|\overrightarrow{P}\|}\right)\right),
\end{aligned}$$

where $\overrightarrow{m}$ is a vector from p perpendicular to $\hat{v}$, $\vec{n}$ is a vector perpendicular to both $\overrightarrow{m}$ and $\hat{v}$ and q_s is the closest distance between p and the path $\vec{v}$.

Based on the idea of windowing from signal processing [31] we define the following functions:

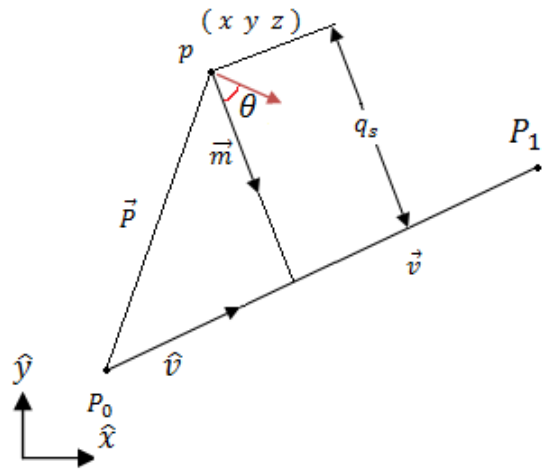


Figure 6.3: Schematics for the straight line vector field deduction.

$$
\begin{aligned}
w_a(w) &= \frac{1}{2}\big(\tanh(a_0 w) - \tanh(a_0(w - \|\overrightarrow{v}\|))\big) \\
w_R(q_s) &= \frac{1}{2}\big(\tanh(a_0(q_s + r_0)) - \tanh(a_0(q_s - r_0))\big) \\
\theta(q_s) &= \frac{\pi}{2} e^{-a_0 q_s^2},
\end{aligned}
$$

where a_0 and r_0 are tunning parameters, $w_a(w)$ is a windowing function along the path, $w_R(q_s)$ is a windowing function perpendicular to the path and $\theta(q_s)$ is a function that regulates the resulting vector field's orientation. Figure 6.4 shows graphs of these functions.

Functions $w_a(w)$ and $w_R(q_s)$ are windowing functions customized for generating a smooth path following vector field, for a path made out of line segments.

We make use of the following transformation [48] that applies to $\overrightarrow{m}$:

$$
M = \begin{bmatrix}
n_x^2 - c\theta(n_x^2 - 1) & n_x n_y - n_z s\theta - n_x n_y c\theta & n_x n_z + n_y s\theta - n_x n_z c\theta & 0 \\
n_x n_y + n_z s\theta - n_x n_y c\theta & n_y^2 - c\theta(n_y^2 - 1) & n_y n_z - n_x s\theta - n_y n_z c\theta & 0 \\
n_x n_z - n_y s\theta - n_x n_z c\theta & n_y n_z + n_x s\theta - n_y n_z c\theta & n_z^2 - c\theta(n_z^2 - 1) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \tag{6.3}
$$

where $[n_x \quad n_y \quad n_z]^T$ is a vector perpendicular to $\overrightarrow{m}$ and $\overrightarrow{P}$, and sines and cosines are abbreviated as s and c respectively.

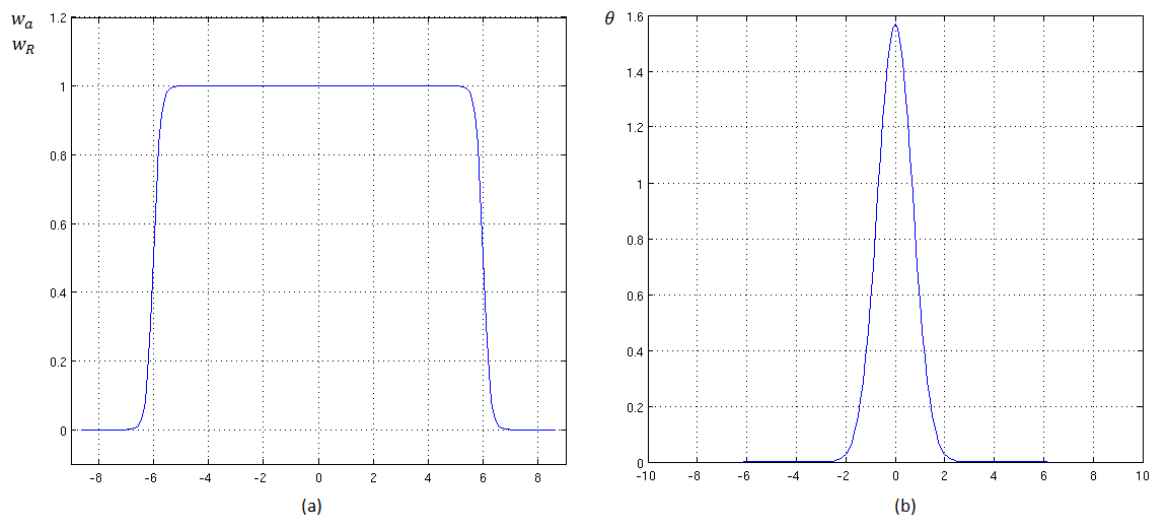


Figure 6.4: Windowing function shape for a) w_a, w_R. and b) θ.

The input vector field $u_{s,i}$ for the ith straight path segment is given by:

$$u_{s,i} = w_{ai}(\overrightarrow{P} \cdot \hat{v}) w_{Ri}(q_s) \frac{(M_i \cdot \overrightarrow{m})}{\|M_i \cdot \overrightarrow{m}\|},$$

and the vector field is shown in Fig. 6.5.

The input that steers the vehicle along any straight line path components is made of the superposition of all individual fields, made possible by the windowing functions:

$$u_s = \sum_{i=1}^{n} u_{s,i}$$

The windowing functions ensure that the straight path segment control law is activated only in the neighborhood of the respective line segments.

6.2 Following curved - Bézier line segments

We match desired curved reference paths using quadratic Bézier curves to define a quadratic Bézier path, we need three points P_0, P_1, and P_2, see Fig. 6.6 for a definition of the quantities of interest with respect to these points, and compare with Fig. 6.3.

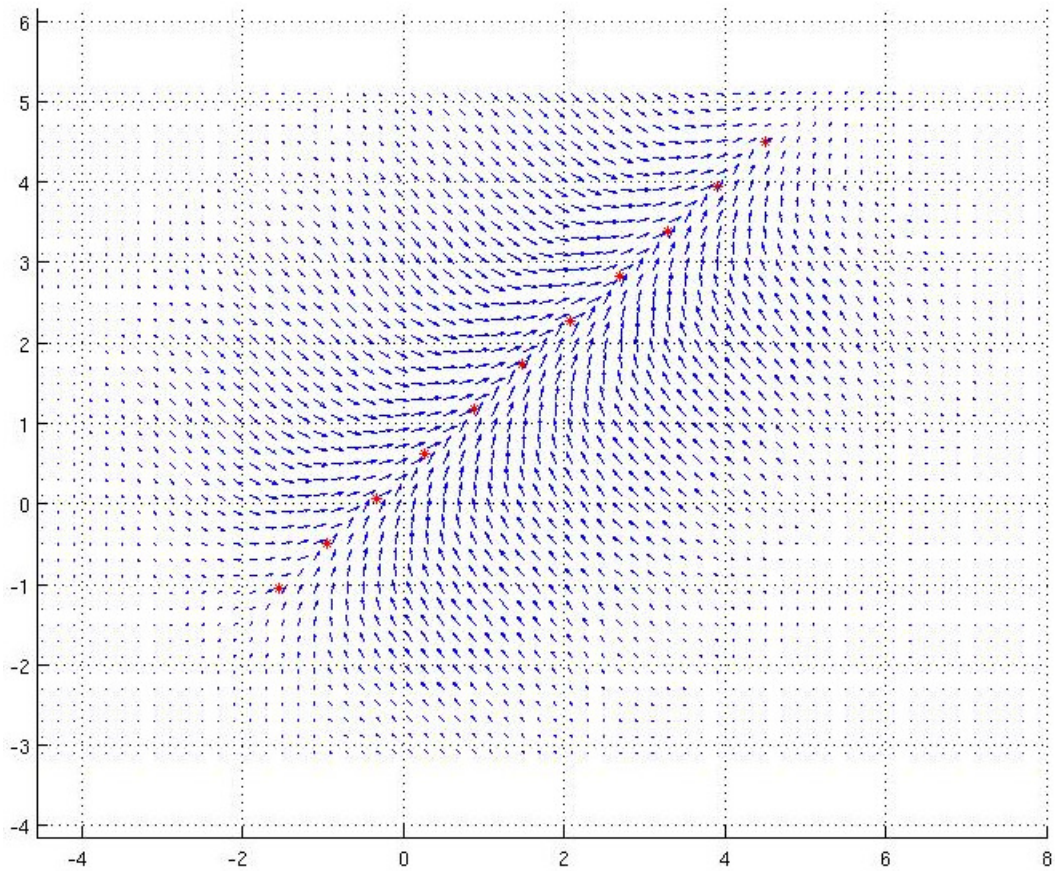


Figure 6.5: Resulting vector field for a straight line. Notice that vectors away from the line segment are attenuated by the windowing functions both along $\vec{v}$ and transversely. A normalization of the vector field is applied after adding up all the path segments to be able to attract the center of mass of the formation from relatively anywhere.

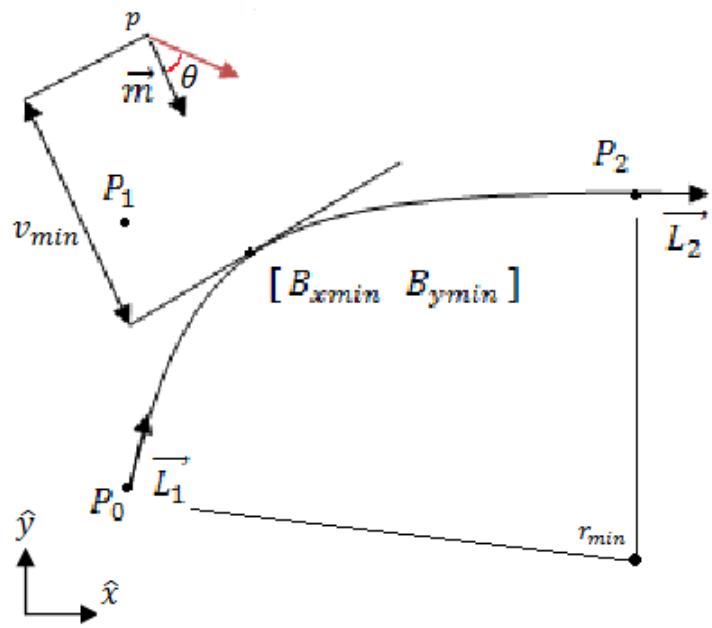


Figure 6.6: Schematics for the Bézier line vector field deduction.

The squematics is traced over an $\mathbb{R}^3$ space defined by the plane made by P_0, P_1, and P_2, $\hat{L}_1$ and $\hat{L}_2$ are unit vectors tangent to the Bézier curve, r_{min} is a common perpendicular point to $\hat{L}_1$ and $\hat{L}_2$, v_{min} is the closest distance from point p to the Bézier curve and $[B_{xmin}B_{ymin}]$ is a point in the Bézier curve which yields v_{min}.

A Bézier curve is given by:

$$B_x = P_{0x}(1-t)^2 + 2P_{1x}t(1-t) + P_{2x}t^2$$
$$B_y = P_{0y}(1-t)^2 + 2P_{1y}t(1-t) + P_{2y}t^2.$$

r_{min} is given by:

$$r_{min} = \begin{bmatrix} \hat{L}_1 & \hat{L}_2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} L_1 \cdot P_0 \\ L_2 \cdot P_2 \end{bmatrix}$$

Now, with vector $\vec{m} = [B_{xmin} - x_p \quad B_{ymin} - y_p]$, we make use of (6.3) to formulate the jth Bézier line segment:

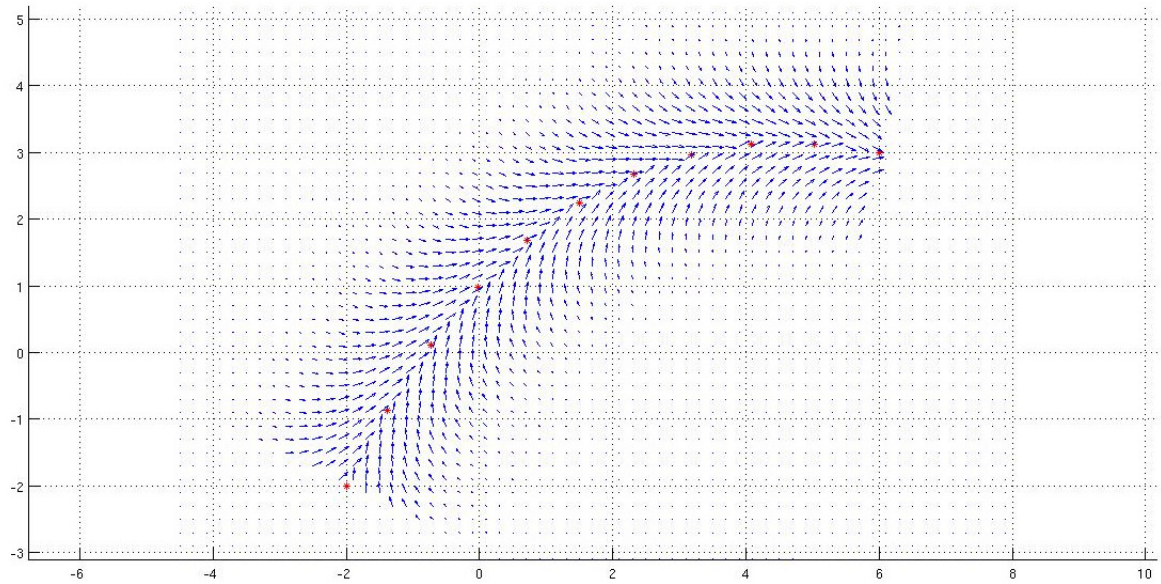


Figure 6.7: Resulting vector field for a Bézier line. Notice again that vectors away from the line segment are attenuated by the windowing functions both tangentially and transversely.

$$u_{b,j} = w_{Ri}(\sqrt{v_{min}})\frac{(M_i \cdot \overrightarrow{m})}{\|M_i \cdot \overrightarrow{m}\|}.$$

The vector field is shown in Fig. 6.7.

The vector field that acts on the vehicle to steer it along any curved path given by:

$$u_b = \sum_{j=1}^{m} u_{b,j},$$

where, again, the windowing functions activate the controller when the formation is in the neighborhood of a Bézier curve, and switch off smoothly all other path following control inputs.

6.3 Convergence Analysis

We provide a convergence analysis for straight-line segments and construct a diffeomorphism that maps the case of Bézier line segments into straight-line ones.

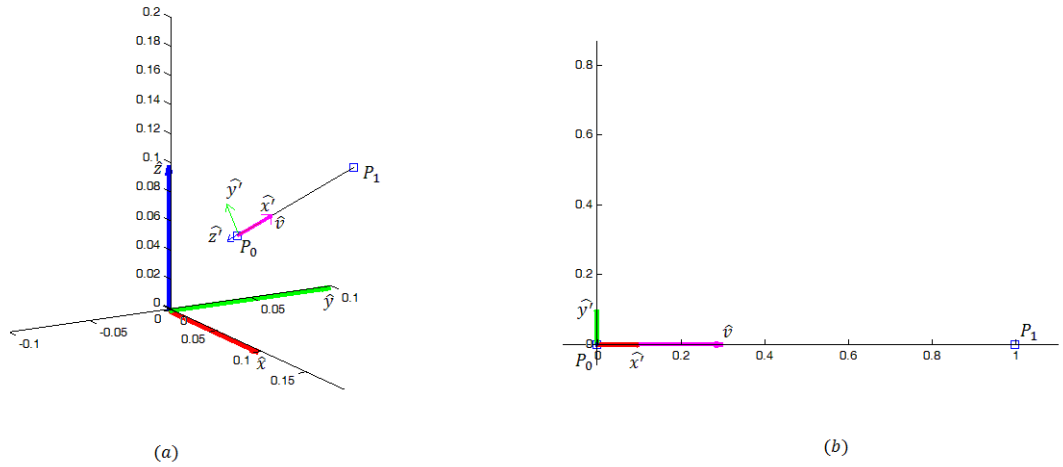


Figure 6.8: Going from a general case on $\mathbb{R}^3$ to a simplified version on $\mathbb{R}^2$. (a) General Case. (b) Simplified Sketch obtained with a translation and rotation matrix

6.3.1 Convergence to straight-line segments

Straight Line Segments 1. *For a point particle we define the dynamics*

$$\dot{x}_{cm} = \sum_{i=1}^{n} u_{s,i}, \quad u_{s,i} = (w_{ai}(\overrightarrow{P} \cdot \hat{v}))(w_{Ri}(q_s))\frac{M_i \cdot \vec{m}}{\|M_i \cdot \vec{m}\|}. \tag{6.4}$$

Inside the domain

$$D = \{x_{cm} \in \mathbb{R}^3 : 0 \leq \vec{v} \cdot x_{cm} \leq 1\},$$

the system converges to the set $A = \{k\vec{v} : 0 \leq k \leq 1\}$

Proof. Without lose of generality take the $\hat{x}$ axis of the reference frame along the reference path A and consider the sketch (b) of Fig. 6.8. (This is a translation-rotation of the original problem where we put the origin at P_0 and $\vec{v} = P_1 - P_0$).

We can write

$$\vec{m} = \hat{v} \times (\hat{v} \times x_{cm}) = (\hat{v} \otimes \hat{v} - I)x_{cm},$$

where I is the identity matrix, $\otimes$ is the outer product. Consider a weak-Lyapunov candidate function

$$V(x_{cm}) = \frac{1}{2}(x_{cm1}^2 + x_{cm2}^2 + x_{cm3}^2).$$

Taking the derivative of this function with respect to time along (6.2)

$$\dot{V}(x_{cm}) = \nabla(V)\dot{x}_{cm} = \frac{(w_a(\overrightarrow{P} \cdot \hat{v}))(w_R(q_s))}{\|M \cdot (\hat{v} \otimes \hat{v} - I)x_{cm}\|}\big[M(\hat{v} \otimes \hat{v} - I)x_{cm}\big]x_{cm} \tag{6.5}$$

The numerator of the scalar factor is composed of windowing functions that take values between zero and one, and the denominator is a scaling factor to make the vector on the right of the expression be unitary, therefore the sign of the right-hand side of (6.5) is determined by

$$\big[M(\hat{v} \otimes \hat{v} - I)x_{cm}\big]x_{cm} = \big[M((\hat{v} \otimes \hat{v})x_{cm} - x_{cm})\big]x_{cm},$$

as the term $\hat{v} = [1 \quad 0 \quad 0]^T$ due to(6.3) and the term $(\hat{v} \otimes \hat{v})x_{cm} - x_{cm}$ becomes

$$(\hat{v} \otimes \hat{v})\vec{x} - \vec{x} = \begin{bmatrix} 0 \\ -x_{cm2} \\ -x_{cm3} \\ 1 \end{bmatrix}$$

and the rotation matrix M becomes:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

then

$$M((\hat{v}\otimes\hat{v})x_{cm}-x_{cm}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -x_{cm2} \\ -x_{cm3} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ -x_{cm2}\cos(\theta)+x_{cm3}\sin(\theta) \\ -x_{cm2}\sin(\theta)-x_{cm3}\cos(\theta) \\ 1 \end{bmatrix}$$

computing the dot product with x_{cm},

$$\begin{bmatrix} 0 \\ -x_{cm2}\cos(\theta)+x_{cm3}\sin(\theta) \\ -x_{cm2}\sin(\theta)-x_{cm3}\cos(\theta) \end{bmatrix}^T \cdot \begin{bmatrix} x_{cm1} \\ x_{cm2} \\ x_{cm3} \end{bmatrix} =$$

$$\begin{aligned} & x_{cm2}x_{cm3}\sin(\theta) - x_{cm2}^2\cos(\theta) \\ & - x_{cm2}x_{cm3}\sin(\theta) - x_{cm3}^2\cos(\theta) \\ & = -\cos(\theta)(x_{cm2}^2 + x_{cm3}^2) \end{aligned}$$

finally, after substitution, we get

$$\dot{V}(x_{cm}) = -\frac{(w_a(\overrightarrow{P}\cdot\hat{v}))(w_R(q_s))}{\|M\cdot(\hat{v}\otimes\hat{v}-I)x_{cm}\|}\big(c\theta(x_{cm2}^2+x_{cm3}^2)\big) \leq 0$$

as $\dot{V}(x_{cm}) \leq 0$ and $V(x_{cm})$ is radially unbounded and continuous the set

$$\Omega_c = \{x_{cm2}, x_{cm3} : V(x_{cm}) \leq c\}, \quad \Omega_c \subset D$$

is a compact, positively invariant set. A set $E = \{x_{cm} \in \Omega_c : \dot{V}(x_{cm}) = 0\}$ is given by $E = \{x_{cm} \in \Omega_c : 0 \leq x_{cm1} \leq 1, x_{cm2} = 0, x_{cm3} = 0\}$.

Vector field (6.4) along set E evaluates as

$$\dot{x}_{cm} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore the set E is an invariant set. From Theorem 4.4 of [18], we conclude that every trajectory starting on Σ_c appoaches set E as $t \to \infty$. Note that $E = A$.

□

Once a trajectory is in E, the dynamics of the formation to be $\dot{x}_{cm} = \hat{v}$, which implies that the system move along the desired path.

6.3.2 Convergence to Bézier line segments

For the convergence analysis, we show there is a diffeomorphism that can map the curved Bézier line into a straight line. In principle, the pull-back map of this diffeomorphism can translate the straight line controller so that it is used for convergence to Bézier curves. A diffeomorphism is merely a coordinate transformation; it does not affect the convergence properties of a system.

The structure of the two controllers is the same, it is merely the scaling that changes; vector $\vec{m}$ always points toward the reference curve along the normal vector to the curve.

First, we start by translating and rotating the locus of the Bézier Line Segment to simplify like the expressions of the Bézier curve and the roots of some polynomials. We denote the initial referential as I_r and the referential of analysis as I_a. Then a point from referential I_r translated to referential I_a is given by

$$\begin{aligned} {}^{r}_{a}p &= \begin{bmatrix} \cos(\vartheta) & \sin(\vartheta) \\ -\sin(\vartheta) & \cos(\vartheta) \end{bmatrix} \left({}_{r}p - {}_{r}P_0\right) \\ \vartheta &= \text{atan2}_{dMia}({}_{r}P_{2y} - {}_{r}P_{0y}, {}_{r}P_{2x} - {}_{r}P_{0x}), \end{aligned}$$

where $\text{atan2}_{dMia}(\cdot,\cdot)$ is a customized function that returns the angle with respect to a reference frame and is defined by

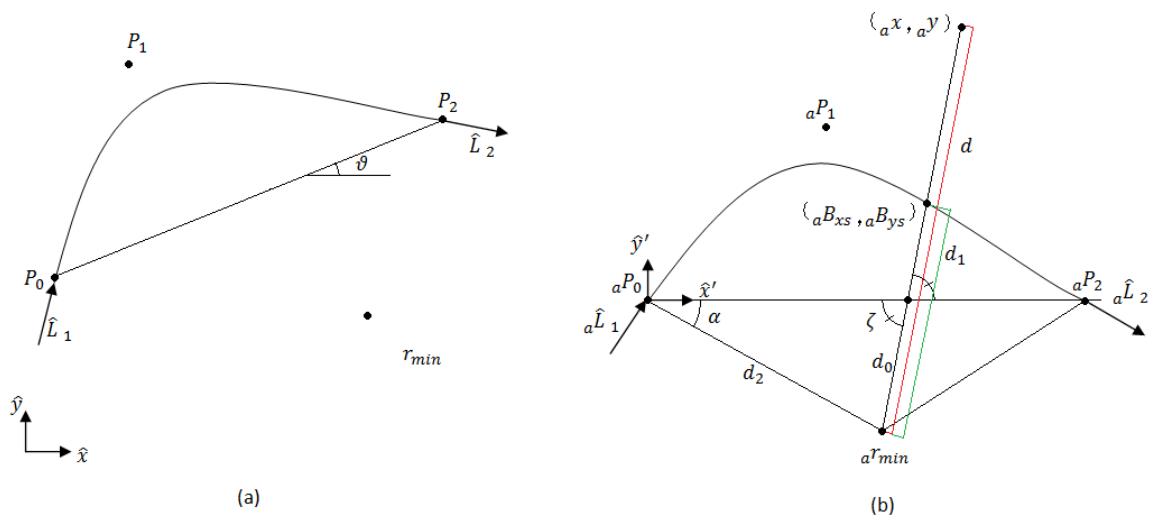


Figure 6.9: Mapping to simplify the analysis for Bézier line segments. (a) General case of a Bézier line path. (b) Simplified version.

$$\text{atan2}_{dMia}(y,x) = \begin{cases} \dfrac{\arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right)\arcsin\left(\frac{y}{\sqrt{x^2+y^2}}\right)}{\sqrt{\arctan\left(\frac{y}{x}\right)^2}} & x \neq 0, y \neq 0 \\ \pi/2 & x = 0, y > 0 \\ -\pi/2 & x = 0, y < 0 \\ \pi & y = 0, x < 0 \\ 0 & y = 0, x > 0 \\ 0 & x = 0, y = 0 \end{cases}$$

This mapping takes the Bézier curve of Fig. 6.9 and maps it to the line interval P_0, P_2 in Fig. 6.9(b).

The Bézier Line expressions become:

$$ {}_aB_x = 2{}_aP_{1x}(1-t)t + {}_aP_{2x}t^2$$

$$ {}_aB_y = 2{}_aP_{1y}(1-t)t.$$

To determine point $({}_aB_{xs}, {}_aB_{ys})$ that is colinear to both ${}_ar_{min}$ and $({}_ax, {}_ay)$, we need to define $\vec{v}_1$ and $\vec{v}_2$ as

$$\vec{v}_1 = \begin{bmatrix} {}_a x - {}_a r_{minx} \\ {}_a y - {}_a r_{miny} \end{bmatrix} \qquad \vec{v}_2 = \begin{bmatrix} B_{xs} - {}_a r_{minx} \\ B_{ys} - {}_a r_{miny} \end{bmatrix}$$

and find the parameter $t_s \in t_{1,2} : 0 \leq t \leq 1$ to satisfy the constrain $\vec{v}_1 \times \vec{v}_2 = 0$. After computing the cross product we arrive to the expression

$$\begin{aligned} &- (2_a P_{1y}({}_a x - {}_a r_{minx}) - 2_a P_{1x}({}_a y - {}_a r_{miny}) + {}_a P_{2x}({}_a y - {}_a r_{miny}))t^2 \\ &+ (2_a P_{1y}({}_a x - {}_a r_{minx}) - 2_a P_{1x}({}_a y - {}_a r_{miny}))t \\ &+ {}_a r_{minx}({}_a y - {}_a r_{miny}) - {}_a r_{miny}({}_a x - {}_a r_{minx}) = 0 \end{aligned}$$

point $({}_a B_{xs}, {}_a B_{ys})$ is given by

$$\begin{aligned} {}_a B_{xs} &= 2_a P_{1x}(1 - t_s)t_s + {}_a P_{2x} t_s^2 \\ {}_a B_{ys} &= 2_a P_{1y}(1 - t_s)t_s \end{aligned}$$

Now d, d_0, d_1 and d_2 are given by

$$\begin{aligned} d_2 &= \sqrt{(r_{minx})^2 + (r_{minx})^2} \\ d_0 &= \frac{\sin\alpha}{\sin\zeta} d_2 \\ d_1 &= \sqrt{({}_a B_{xs} - {}_a r_{minx})^2 + ({}_a B_{ys} - {}_a r_{miny})^2} \\ d &= \sqrt{({}_a x - {}_a r_{minx})^2 + ({}_a y - {}_a r_{miny})^2} \\ \alpha &= |\text{atan2}_{dMia}({}_a r_{miny}, {}_a r_{minx})| \\ \zeta &= |\text{atan2}_{dMia}({}_a y - {}_a r_{miny}, {}_a x - {}_a r_{minx})| \end{aligned}$$

Finally, the diffeomorphism which transforms the Bézier curve into a straight line is given by

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \frac{d_0 d_1}{d^2} \begin{bmatrix} {}_a x - {}_a r_{minx} \\ {}_a y - {}_a r_{miny}. \end{bmatrix} \tag{6.6}$$

We can verify that (6.6) maps the controller expression for the straight line segment into the Bzier line segment controller.

Chapter 7

FLOCKING CONTROL

First, on Section 7.1 we present the formal definition used to prove that our framework achieves flocking. Then on Section 7.2 and Section 7.3 we explain how the states for the flocking with formation control closed loop system are obtained and show how this mechanism to obtain relative position among the robots avoids interference between the control laws for each objective (flocking, formation control, and path following) respectively. Finally, on Section 7.4, we present the mathematical proof that states that our methodology achieves flocking with formation control.

7.1 Definition

Definition 9 (Flocking)**.** *Flocking is the behavior exhibited by a group of N particles for which all pairwise velocity differences converge asymptotically to zero, collisions between the particles are avoided, and the particles are distributed around a single moving point.*

7.2 From Absolute to Relative Position

Consider the incidence matrix B of the complete graph having the nodes of the formation graph of Definition 1. As heads and tails of the edges are marked with 1 and -1 respectively, we can multiply the concatenated vector of robots position to get relative positioning:

$$q = Bp. \tag{7.1}$$

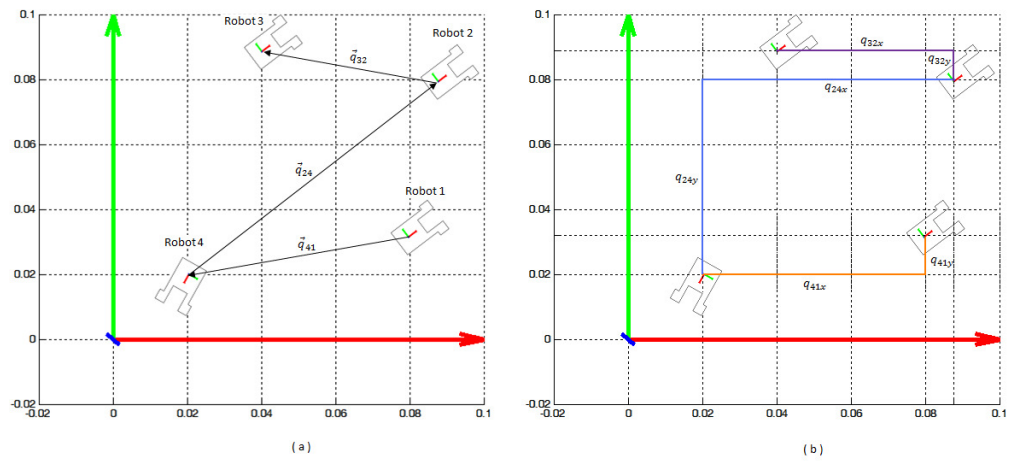


Figure 7.1: Schematics to show the non curvilinear mapping due to (7.1). (a) Example of q_{ij} vectors for four robots. (b) Decomposition of q_{ij} vectors showing their rectangular nature.

where a single component is seen as:

$$q_{ij} = p_i - p_j \tag{7.2}$$

which implies that robot i is head and robot j is tail of and edge in the formation graph (see Definition 1). For this reason, even if all results are stated here with respect to a relative position vector q that includes all relative positions, it will be understood that in an actual implementation only the information from $N-1$ pairs of robots is used to compute the proposed control laws.

Note that (7.1) is a mapping from a non curvilinear space to another curvilinear space as shown in Fig. 7.1. Therefore we can apply operatiors such as ∇ as if it were on the Cartesian frame, only with more variables.

7.3 Using Graphs to Capture Formation Topology

To ensure collision avoidance between all agents in [42] the formation graph needs to be taken as the complete graph, because it is implicitly assumed that agents that are not linked by an edge do not communincate and are unaware of each other's

existence. Other approaches like [44] and [12] claim that the only way in which a fixed topology is able to guarantee collision avoidance is when all agents are interconnected to each other.

Note that the path following control has no effect on the dynamics of relative positions because of the decoupling performed by (7.1). Consider the derived dynamics (3.5) for a single robot and the second derivative of (7.2) as

$$\ddot{q}_{ij} = \ddot{p}_i - \ddot{p}_j \quad \Longrightarrow \quad \dot{x}_1 = \Gamma_i - \Gamma_j,$$

and when the same acceleration input Γ_{input} due to the path following control is applied to all the robots involved in the formation, each individual mapping done by (7.1) gives

$$\begin{aligned}
\dot{x}_{1ij} &= (\Gamma_i + \Gamma_{input}) - (\Gamma_j + \Gamma_{input}) \\
\dot{x}_{1ij} &= \Gamma_i - \Gamma_j + \Gamma_{input} - \Gamma_{input} \\
\dot{x}_{1ij} &= \Gamma_i - \Gamma_j.
\end{aligned}$$

Transformation (7.1) works like a noise cancellation circuit based on operational amplifiers: it rejects the common component that is present between two different points of measure; in this case, it cancel out the common input T_{input}.

7.4 Flocking Analysis

Theorem 2 guarantees that we can apply (5.2) as input to a group of robots to have flocking with a desired formation specification.

Theorem 2 (Main Result)**.** *For a group of robots with dynamics*

$$\begin{aligned}
\dot{x}_1 &= u_f \\
\dot{x}_2 &= x_1
\end{aligned} \tag{7.3}$$

x_1 and x_2 are the relative velocity and position stack vectors measured with respect to a desired relative position configuration, i.e., $x_2 = q - c$, respectively, of all the robots involved in the group defined over a workspace $\mathcal{Q}$

$$\mathcal{Q} = \mathbb{R}^{nN(N-1)/2} \times \mathbb{R}^{nN(N-1)/2} - \{(x_1, x_2) : \|x_{2,i}\| < d_0, \forall (i,j) \in N \times N\} \tag{7.4}$$

and u_f is a control input defined as

$$u_f = -\mathrm{Ln}(\nabla(\varphi(x_2))) - x_1$$

where

$$\varphi(x_2) = \frac{\gamma(x_2)}{\beta(d(x_2))}; \qquad d(x_2) = min\{\|x_{2,i}\|\}.$$

Then, collisions among agents are avoided, all agents' velocity vectors become asymptotically the same and the agents' relative position vectors $x_{2,i}$ converge to desired constant vectors defined by the formation specifications.

Proof. The equilibrium of the system is

$$x_1 = 0$$
$$\nabla(\varphi(x_2)) = 0 \Rightarrow \varphi(x_2) = 0$$

and attained when the robots are either at their desired relative configurations or at saddle points, which are isolated as shown in [42]. The attraction region of all unstable equilibria (at saddles) is of measure zero [42]. For this reason, the only practically attainable equilibrium is $x_1 = 0, x_2 = c$. Our weak-Lyapunov function candidate is:

$$V(x_1, x_2) = \frac{1}{2}\|x_1\|_2^2 + \varphi(x_2)$$

which is positive definite because the potential function $\varphi(x_2)$ is positive definite with respect to x_2[1]. At configurations where the minimum relative distance function $d(x_2)$ is nondifferentiable, $\dot{x}$ belongs to a Filippov set (see (4.2)):

$$\dot{x} \in F(x) \tag{7.5}$$

specifically,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \in \begin{bmatrix} -\partial\varphi(x_2) - x_1 \\ x_1 \end{bmatrix} = F(x)$$

The generalized derivative of V along a vector in $F(x)$ is given by ([6, Proposition 2.1.2])

$$V^\circ(x, \xi) = \max\{p^T \cdot \xi, p \in \partial V(x), \xi \in F(x)\}$$

where $\partial V(x)$ is

$$\partial V(x) = \begin{bmatrix} x_1 \\ \partial\varphi(x_2) \end{bmatrix};$$

and

$$\partial\varphi(x_2) = \frac{1}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d) - \gamma_d^\kappa\left(\frac{\partial\beta}{\partial d}\right)\lambda\right), \quad \lambda \in \partial d.$$

[1] φ might depend on a subset of the components of x_2 but because of the $N-1$ connectivity (graph) constrains between the agents, determination of this subset, determines the whole set.

Hence

$$
\begin{aligned}
V^{\circ}(x,\zeta) = \max\Bigg\{ & -\frac{x_1^T}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d)^T - \gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)\zeta\right) - \|x_1\|_2^2 \\
& + \frac{x_1^T}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d)^T - \gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)\xi\right) \quad : \quad \zeta,\xi \in \partial d\Bigg\} \Leftrightarrow \\
V^{\circ}(x,\zeta) = \max\Bigg\{ & \frac{1}{\beta^2}\left(\gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)x_1^T\cdot\zeta\right) - \|x_1\|_2^2 - \frac{1}{\beta^2}\left(\gamma_d^{\kappa}\Big(\frac{\partial\beta}{\partial d}\Big)x_1^T\cdot\xi\right) \\
& : \quad \zeta,\xi \in \partial d\Bigg\} \Leftrightarrow \\
V^{\circ}(x,\zeta) = \max\Bigg\{ & -\|x_1\|_2^2 + \left(\frac{\gamma_d^{\kappa}}{\beta^2}\Big(\frac{\partial\beta}{\partial d}\Big)x_1^T\cdot(\zeta-\xi)\right) \quad : \quad \zeta,\xi \in \partial d\Bigg\} \Leftrightarrow \\
V^{\circ}(x,\zeta) = \max\Bigg\{ & -\|x_1\|_2^2 + \left(\frac{\gamma_d^{\kappa}}{\beta^2}\Big(\frac{\partial\beta}{\partial d}\Big)\|x_1\|\|\zeta-\xi\|\right) \quad : \quad \zeta,\xi \in \partial d\Bigg\}.
\end{aligned}
$$

Since the maximum difference $\|\zeta-\xi\|$ between vectors in ∂d is attained when ζ and ξ are unit vectors along different coordinate directions, it follows

$$
V^{\circ}(x,\zeta) \leq -\|x_1\|_2^2 + \sqrt{2}\frac{K_\zeta}{\beta}\|x_1\|\varphi \tag{7.6}
$$

where K_ζ is an upper bound on $\frac{\partial\beta}{\partial d}$, see [42]. Case I: suppose φ is differentiable at x_2 , then $\zeta = \xi$ and $V^{\circ}(x,\zeta) = \dot{V}(x)$

$$
\dot{V}(x_1,x_2) = -\|x_1\|^2 \leq 0.
$$

Since $V(x_1,x_2)$ is radially unbounded, the set

$$
\Omega = \{(x_1,x_2) \in \mathbb{R}^{nN(N-1)} | V(x_1,x_2) \leq \text{Constant}\}
$$

is a positive invariant set that is closed by continuity and bounded because V is radially unbounded. As $\Omega = \{(x_1,x_2) : V(x_1,x_2) \leq \text{Constant}\}$ is a positive invariant set, solutions in Ω cannot escape Ω and therefore robots cannot go to a collision configuration because these configurations occur where $V \to \infty$.

Define $E \subset \Omega$ as follows:

$$
\begin{aligned}
& E = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | \quad \dot{V}(x_1, x_2) = 0\} \\
\Leftrightarrow \quad & E = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | \quad x_1 = 0, \varphi(x_2) = 0\} \\
\Leftrightarrow \quad & E = \{(x_1, x_2) \in \mathbb{R}^{nN(N-1)} | \quad x_1 = 0, x_2 = c\}.
\end{aligned}
$$

The dynamics on E is:

$$0 = -\nabla(\varphi(c)) - x_1$$

for which the biggest invariant set $M \subset E$ is $(x_1, x_2) = (0, c)$. So the trivial solution is the only one which can stay in E.

Then by LaSalle principle [18] we know that every solution starting in Ω will approach M as $t \to \infty$. By Corollary 4.1 in [18] we know that the equilibrium point is asymptotically stable.

Case II: suppose φ is nondifferentiable at x_2, i.e. $\zeta \neq \xi$; then, we have regions of the state space where $V^\circ(x, \zeta)$ is positive (See Fig . 7.2). However the solutions for (7.3) are bounded. Then, for every constant $M \geq 0$, there exits some $t \geq 0$ such that $V(x(t)) > M$. Pick $x(0)$ at a point where $V^\circ(x, \zeta) \geq 0$ which means that

$$\|x_1\| < \frac{\varphi}{\beta}\|\xi - \zeta\| K_\zeta \tag{7.7}$$

is the region on the upper left hand side in both parts of Fig. 7.2. Now if $x(t)$ is growing without bound it will end up hitting the boundary of the workspace of φ. From [42] we know that $-\nabla\varphi$ is unbounded pointing out towards the direction that x_2 decreases. As x_2 has to remain bounded, the only option left for $V(x)$ to increase is if x_1 increases; but that contradicts (7.7). Therefore neither x_1 nor x_2 can grow without bound. Lemma 4.8 of [23] implies that if $x(t)$ is a bounded solution and has no positive limit point on the boundary of the workspace, then $x(t)$ is precompact.

Define ([36, Theorem 2.11.]):

$$
\begin{aligned}
\nu : G \to \mathbb{R}, \quad x \to \nu(x) &:= \max\{V^\circ(x, \zeta) | \zeta \in F(x)\} \\
&\leq -\|x_1\|^2 + \sqrt{2}\frac{K_\zeta}{\beta}\|x_1\|\varphi
\end{aligned}
$$

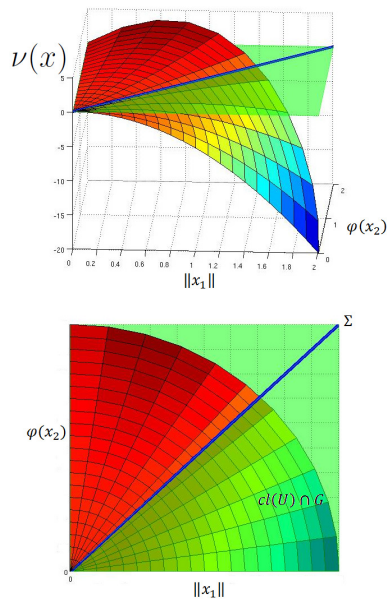


Figure 7.2: Sketch for $V^\circ(x,\zeta)$. $V^\circ(x,\zeta)$ has three important regions, $V^\circ(x,\zeta) \geq 0$, $V^\circ(x,\zeta) \leq 0$ and $V^\circ(x,\zeta) = 0$.

There is a set $U \subset \mathcal{Q}$ that is not empty such that $u(x) \leq 0$:

$$U = \left\{ \|x_1\| \geq \frac{\varphi}{\log(\mu)} \|\xi - \zeta\| K_\zeta \right\}, \tag{7.8}$$

where μ is a constant. For the next part we need Theorem 3 (see Chapter4) and Theorem 4 from [36] and [6] respectively:

Theorem 3 (Ryan's Invariance Principle [36])**.** *Let $V : \mathcal{Q} \to \mathbb{R}$ be locally Lipschitz. Define*

$$\nu : \mathcal{Q} \to \mathbb{R}, \quad x \to \nu(x) := \max\{V^\circ(x,\zeta) | \zeta \in F(x)\}$$

Suppose that $U \subset \mathcal{Q}$ is non-empty and that $\nu(x) \leq 0$ for all $x \in U$. If x is a precompact solution of (7.5) with trajectory in U, then for some constant $k \in V(\mathrm{cl}(U) \cap \mathcal{Q})$, x approaches the largest weakly-invariant set in $\Sigma \cap V^{-1}(k)$, where

$$\Sigma = \{x \in cl(U) \cap \mathcal{Q} | \nu(x) \geq 0\}.$$

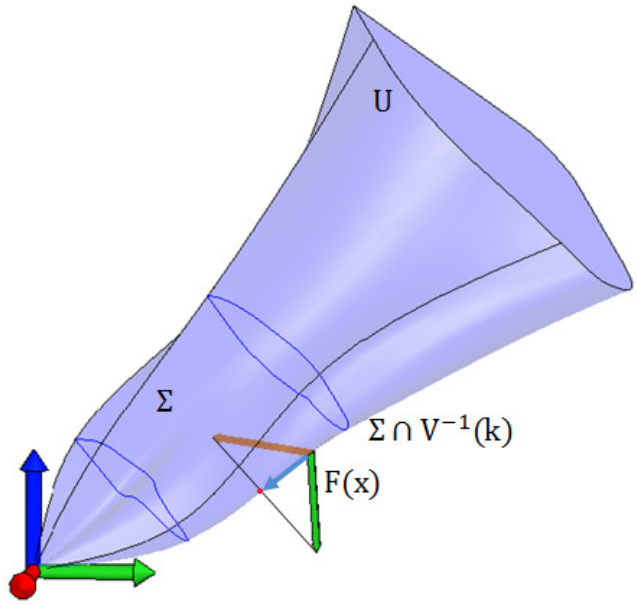


Figure 7.3: Filippov set along $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$.

System (7.3) satisfies all the conditions of the previous theorem. We know that the trajectory of x will be in U (see the argument related to the boundedness of (7.7)). Because of Theorem 1, we know that the trajectory of x will converge to the largest weakly-invariant set in $\Sigma \cap V^{-1}(c)$. What it is left for us to prove is that once in the set $\Sigma \cap V^{-1}(c)$, the system vector field vanishes only at $[x_1 \quad x_2]^T = [0 \quad c]^T$ and it remains tangent to the set $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$.

The Filippov set (7.5) along $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$, see Fig. 7.3, is given by:

$$F(x) = \begin{bmatrix} -\frac{1}{\beta^2}\left(\kappa\beta(\gamma_d)^{\kappa-1}\nabla(\gamma_d) - \gamma_d^{\kappa}\left(\frac{\partial\beta}{\partial d}\right)\lambda\right) - x_1 \\ x_1 \end{bmatrix}; \qquad 0 \notin F(x)$$

Where $\lambda \in \partial d(\|x_2\|)$, the convex hull of unit vectors e_i and any $\lambda \in \partial d(\|x_2\|)$ can be wirtten as $\lambda = \Sigma_i a_i e_i$ where $\Sigma_i a_i = 1$; thus λ cannot be zero. On the other hand, the only place in which x_1 and γ_d are zero, and therefore $F(x) = 0$, is only at

$x_1 = 0, \varphi(c) = 0$ by construction of $\varphi(x_2)$. For this to happen, $\nabla(\gamma_d) = x_2 - c$ so the only place it is zero is when $x_2 = c$ and in the event $\frac{\partial \beta}{\partial d} = 0$, $\beta =$ constant.

Since $F(x)$ contains a tangent direction to the set $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$, the set $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$ cannot be weakly-invariant, rather, there is a solution that escapes any bounded proper subset of $\Sigma \setminus \{x | x_1 = 0, \varphi(c) = 0\}$. Let

$$d_\Sigma(x) = \inf\{\|x - \sigma\| : \sigma \in \Sigma\}.$$

The tangent cone T_Σ to cl(Σ) at a point x is defined as:

$$T_\Sigma(x) = \{\zeta : d^\circ(x, \zeta) = 0\}.$$

Algebraically, the fact that some $v \in F(x)$ is in this tangent cone can be expressed as

$$\exists v \in F(x) : \max\{< \zeta, v >: \zeta \in d_\Sigma(x)\} = 0 \quad \Leftrightarrow \quad \exists v \in F(x) : \quad < \zeta, v > \leq 0; \quad (7.9)$$

Note that computing $d_\Sigma(x)$ requires the use of a point σ in the closure of Σ that gives orthogonality and which we do not know; but the next theorem enable us to use an alternative expression for the normal of Σ.

Theorem 4 ([6]). *Let x belong to $cl(\Upsilon)$. Then $\partial d_\Upsilon(x)$ equals the convex hull of the origin and the set*

$$\left\{ w = \lim \frac{w_i}{\|w_i\|} : w_i \perp \Upsilon \quad at \quad x_i \to x, w_i \to 0 \right\}$$

To determine the sign in (7.9) we need the direction of vectors, not their magnitude. It is convenient to scale an arbitrary element $v \in F(x)$ as

$$\frac{v}{\|x_1\|} = \begin{bmatrix} -\frac{1}{\beta\|x_1\|}\left(2\kappa(\gamma_d)^{\kappa-1}(x_2 - c) - \frac{\beta}{\sqrt{2}}\|x_1\|\left(\sum_i a_i \xi_i\right)\right) - \frac{x_1}{\|x_1\|} \\ \frac{x_1}{\|x_1\|} \end{bmatrix}$$

since

$$\Sigma = \left\{ x : \|x_1\|^2 - \frac{2}{\beta^2} K_\zeta^2 \varphi(x_2)^2 := \Psi(x) = 0 \right\}$$

the set of vectors normal to Σ is given as

$$\frac{\partial \Psi(x)}{\|x_1\|} = \begin{bmatrix} 2\frac{x_1}{\|x_1\|} \\ -2\frac{\|x_1\|}{\beta K_\zeta}\left(\sum_i a_i \xi_i\right) \end{bmatrix}$$

then ζ in (7.9) is in $\frac{\partial \Psi(x)}{\|x_1\|}$ (magnitude is irrelevant for (7.9); it is the direction that matters). Therefore we would need at least one element of $F(x)$ to guarantee the condition in (7.9), recall that $F(x)$ is a subset of $T_\Sigma(x)$ and write

$$\begin{aligned} \zeta^T \cdot \frac{v}{\|x_1\|} &= -\frac{4\kappa(\gamma_d^{\kappa-1})}{\beta\|x_1\|^2} x_1^T \cdot (x_2 - c) + \frac{\sqrt{2}}{\|x_1\|} x_1^T \cdot \left(\sum_i a_i \xi_i\right) - 2 - \frac{2}{K_\xi \beta} x_1^T \cdot \left(\sum_i a_i \xi_i\right) \\ &= -\frac{4\kappa(\gamma_d^{\kappa-1})}{\beta\|x_1\|^2} x_1^T \cdot (x_2 - c) - \frac{2}{K_\xi \beta} x_1^T \cdot \left(\sum_i a_i \xi_i\right) + \frac{\sqrt{2}}{\|x_1\|} x_1^T \cdot \left(\sum_i a_i \xi_i\right) - 2 \end{aligned} \tag{7.10}$$

When going close to $x_1 = 0$, $\varphi(c) = 0 \Rightarrow \gamma_d = 0$ and with an appropiate large exponent κ we can make the first two terms of the last equation in (7.10) arbitrarilly small. Thus

$$\zeta^T \cdot \frac{v}{\|x_1\|} \text{ converges to } \frac{\sqrt{2}}{\|x_1\|} x_1^T \cdot \left(\sum_i a_i \xi_i\right) - 2 \text{ from below, and we know that}$$

$$\frac{\sqrt{2}}{\|x_1\|} x_1^T \cdot \left(\sum_i a_i \xi_i\right) - 2 \leq \sqrt{2} - 2 < 0$$

because ξ_i are unit vectors and $\sum_i a_i = 1$. By continuity of the right hand side of (7.10) and Ψ the largest invariant set in Σ, where $\Psi = 0$ can only be at $x_1 = 0$, $\varphi(c) = 0$.

□

Chapter 8

INTEGRATED SYSTEM

While on the theoretical side we demostrated the correctness of our control law to achieve flocking with formation control and path following, on the practical side we need to verify the applicability of the control law and the degree to which our assumptions are reasonable by performing simulations and experiments. For the experiments (see Chapter 9), we need the integration of several components to give rise to the complete control structure of our formulation. The description of those components and their integration are the main purpose of this chapter.

8.1 System Components

All the components needed to implement the entire methodology are shown in Fig. 8.1. We identify two major layers: a physical hardware layer and a software layer. Note that the connection between the two layers is done between the robot class and the physical mobile robot.

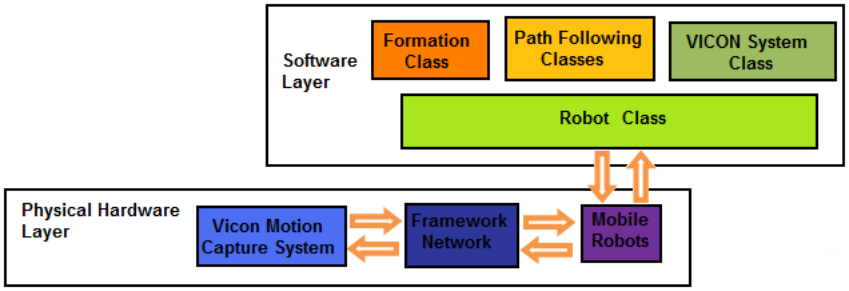


Figure 8.1: Layer hierarchy of the framework.

8.1.1 Physical Hardware Layer

The physical hardware layer is composed by the mobile robots, the VICON motion capture system and the communication network.

8.1.1.1 Mobile Robots

We employ corobots, which are skid-steering differentially driven platforms manufactured by CoroWare Inc. Each corobot has a metallic structure that serves as a platform for any sensor we want to attach to the robots, like a camera. We also attach gray reflective markers to the robots so that the VICON motion capture system is able to monitor their position (Fig. 8.2). Each robot has a Mini-ITX Motherboard with VIA C7 1.5GHz Processor, 1 GB of RAM memory, a wireless communication card and a hard disc of 60 GB that allow each robot to run Ubuntu Maverick Meerkat. Our client programs make the robots move by communicating with PLAYER 3.02, which is a server program that stands between the user codes and the low level controllers of the robot. PLAYER 3.02 is compatible with client programs written in *C*, *C++* or *python*.

8.1.1.2 VICON Motion Capture System

The motion capture system serves as our feedback position sensor. It consists of eight infrared cameras distributed around the laboratory and connected to a dedicated network switch interfaced to a computer, the VICON (system) server. The VICON server has a virtual model (a subject) of each robot in the lab so it can be able to distinguish a particular robots among the group. The VICON server returns the location of a particular robot by executing an algorithm that triangulates the position of the gray markers attached to the robot. Additionally, the VICON server is capable of reporting data at a rate of 100 Hz. The robots position can be saved in a text file.

8.1.1.3 Networking

The network in charge of providing feedback is composed by all the robots, a wireless router and the VICON motion capture system, see Fig. 8.3.

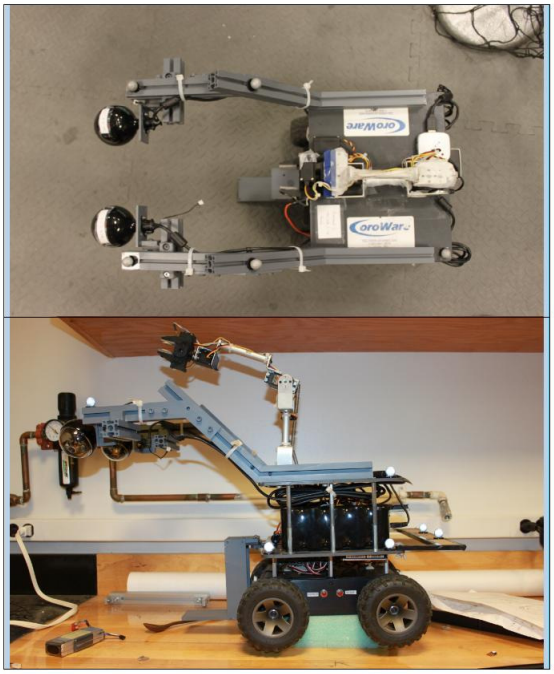


Figure 8.2: Corobot. Differential robot manufactured by CoroWare. The metallic frame is custom made by us for the purpose of data collection.

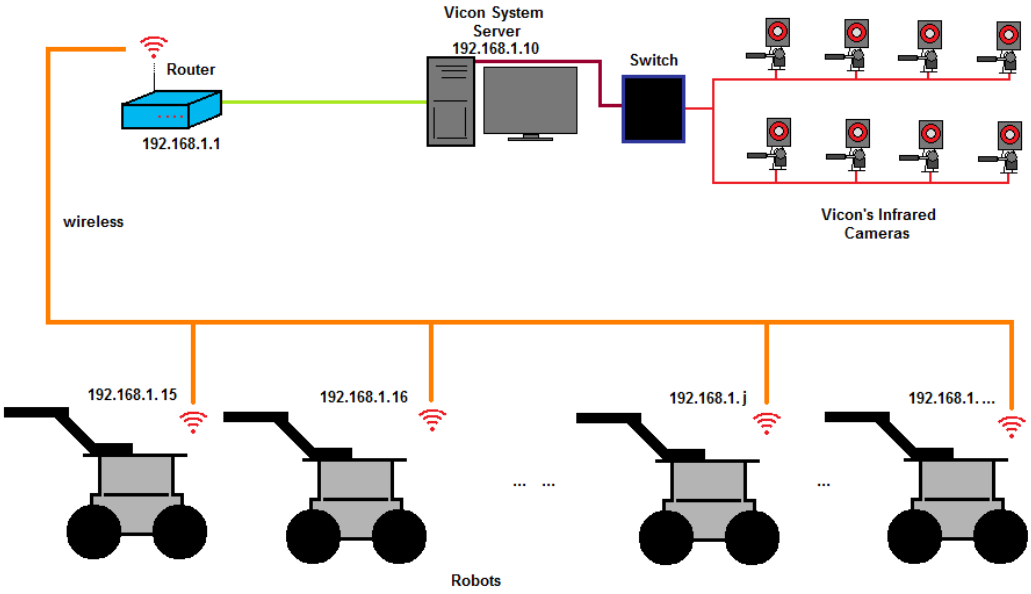


Figure 8.3: Network Schematics. The only practically centralized task is performed in the VICON system server which reports the positioning of the robots in the lab.

The delay reported when executing a ping signal from the VICON Motion Capture System to the robots is around 15 ms, and a total approximate delay of 30 ms due to network traffic, the overal latency of the localization system is not critical for our purposes and can be neglected. The robots and the sensor system communicate through a wireless router with a range that extends beyond the area of our laboratory so the communication signal range is not an issue. The VICON Motion Capture System is capable of reporting the absolute positions of all the robots.

8.1.2 Software Layer

We adopt an object-oriented programming methodology for the implementation of the software layer because its modularity allows the inclussion of different kinds of objects in the same project and the interaction-communication among them; it also allows the existance of several objects that inherits the attributes and methods specified in classes. The Software layer is composed by the following classes: formation software, path following, Vicon system and Robot.

8.1.2.1 Formation software class

This class embeds the formation control algorithm. It executes tasks such as the calculation of the gradient of the navigation function under the space of selected relative positions, the generation of the incidence matrix[1] and its generalized inverse, the update of the relative position vector x_2 and the estimation of the mass center of the formation and the estimation of the error in formation.

8.1.2.2 Path following classes

The two path following classes are the *Path Segment Linear class* and the *Curved Segment class*. They contain the calculation routines to generate the vector fields for straight line and Bézier line path segments, respectively. See Chapter 6.

[1] Although the theory suggests the use of the complete graph's incidence matrix, one can implement the control using a spanning tree instead and exploit the associated computational benefits.

8.1.2.3 Vicon System class

This class is in charge of making the request for updated data to the VICON server. It is composed of an *update* method and a method called `find_sub`; `find_sub` is in charge of collecting position data for a particular robot from an array that contains the information of all the subjects the VICON server is monitoring.

8.1.2.4 Robot class

The robot class contains objects from the previous classes and it is where the interaction between them take place. Note that the previous classes provide the information the robot needs to move so it is reasonable to locate the interaction of all the classes inside a method of the robot class. It is in charge of translating the robot's position and orientation to the linearized dynamics model of (3.5) and map the generated control input back to the nonholonomic robot model. The Robot class is at the same time a client program that communicates with the Player server on the robot.

8.2 Control Architecture

The overall control architecture is shown in Fig. 8.4. Under this architecture, the formation control and flocking algorithms are carried out by the inner loop and the path following algorithm is performed by the outer loop. The position sensor block represents the VICON motion capture system and the Vicon system class, the input-output linearization and dynamics blocks represent the robot class and the group of mobile robots, repectively; the matrix generator, the formation mass center and the u_f blocks represent the formation class. The path following classes are represented by the u_p block.

Note that the inputs for the control signal u_f are B, C and x_2 only; this is because the position sensor block does not return the velocity of the subject it monitors. In order to circumvent this, we formulated a velocity estimator that divides the change of the relative positions by the sampling time.

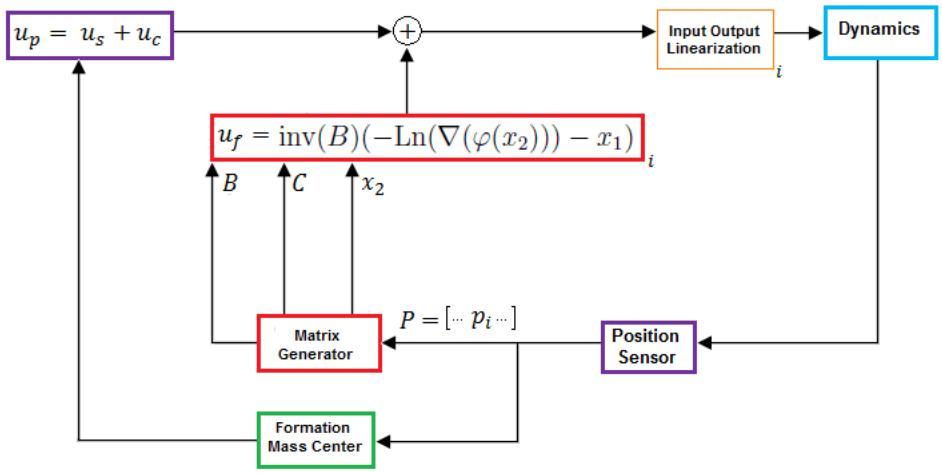


Figure 8.4: Control Architecture. Formation control and path following are integrated in such a way we can guarantee flocking.

The complete control law expressing the integration of path following, formation control and flocking is given by (8.2). The global control signal is present at the output of the only summation symbol shown in the architecture.

$$u_{global} = u_p + \mathrm{inv}(B)(-\mathrm{Ln}(\nabla(\varphi(x_2))) - x_1)$$

where u_p is the path following control contribution, $-\nabla(\varphi(x_2))$ is the formation control contribution and the damping term $-x_1$ is the flocking control contribution and $\mathrm{inv}(B)$ is the Moore-Penrose generalized inverse. Note that this signal is later mapped into the corresponding linear and angular velocity input of each differential mobile robot by the input-output linearization block.

Chapter 9

TESTING AND PERFORMANCE EVALUATION

We complete the testing and performance evaluation of our framework by means of two simulations and two experiments.

9.1 Simulations

For the simulations performed, we designed the path shown in Fig. 9.1. It is composed by three straight lines and two Bézier curves, see Chapter 6. Note that the vector field composed by the five path segments mentioned before produce a smooth vector field directed along the designed route.

Our first simulation is displayed in Fig. 9.2, 9.3 and 9.4. In Fig. 9.2(a), the robots are initially positioned in a vertical straight line and the goal formation is a diamond shape with a robot in the middle; then, during Fig. 9.2(b) and Fig. 9.2(c) robots start falling into the formation and at the same time they are moving in the direction indicated by the path following vector field. By snapshot Fig. 9.2(d) the robots have already achieved the formation specification and the group of robots is on the route. From the snapshot of Fig. 9.3(e) to that of Fig. 9.4(j) we see a steady state behavior in which the formation moves over the designed route without changing its orientation. The group of robots have achieved flocking.

For our second simulation, we have prepared a more challenging configuration by locating the robots around the workspace. Figures 9.5, 9.6 and 9.7 show the snapshots for the second simulation; the formation specification is the same one employed on the first simulation. This time, it takes some time until the snapshot of Fig. 9.7(j) is taken to achieve the formation specification. Again, we want to point out that the robots were converging to the designed route when trying to carry out the formation

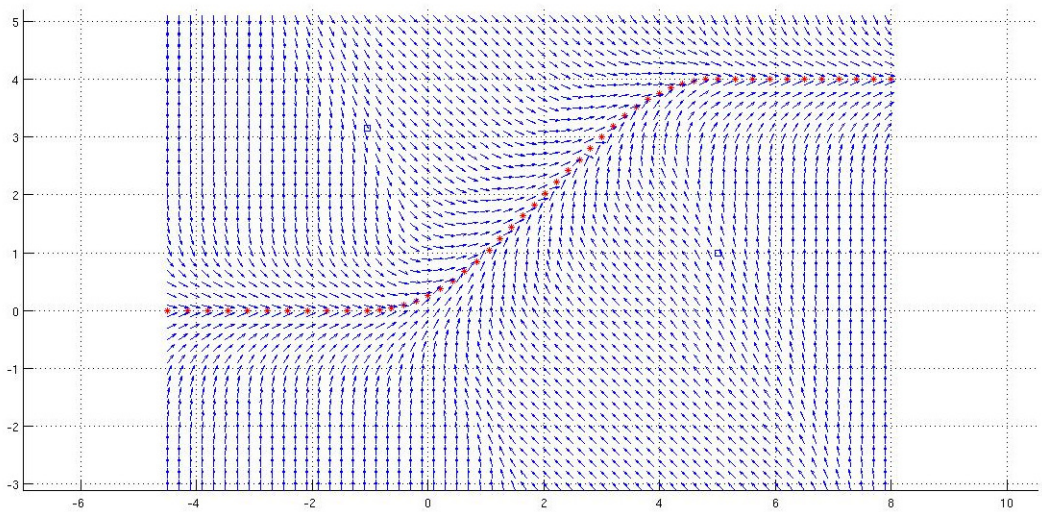


Figure 9.1: Vector field for simulated path. This is the vector field in which our two simulation take place, notice that even when we add up different vector fields, there is no place for dead zones or sinks.

specification, that means both formation control and path following algorithms were executed at the same time. Also note that some of the robots that started at the bottom of the workspace occupy places at the top of the formation configuration and viceversa. For instance, the cian-colored robot started at the very top and ended up at the bottom of the formation, so that robot traversed on a path in which other robots were moving and despite of that, it did not collide.

9.2 Experiments

Our experimental objective is to deploy a group of robots to perform an area search of our laboratory. Each robot has two cameras attached to them to take pictures of the ground; the robots make a formation of a straight line that is perpendicular to the movement in order to create a big scanner covering the ground. It is important for the robots' formation to be compact enough to avoid leaving any space uncovered. However, the constrains in space restricts us to work with two robots and move around a relative small area rather than a long distance.

An important assumption related to the robot platform has to do with the

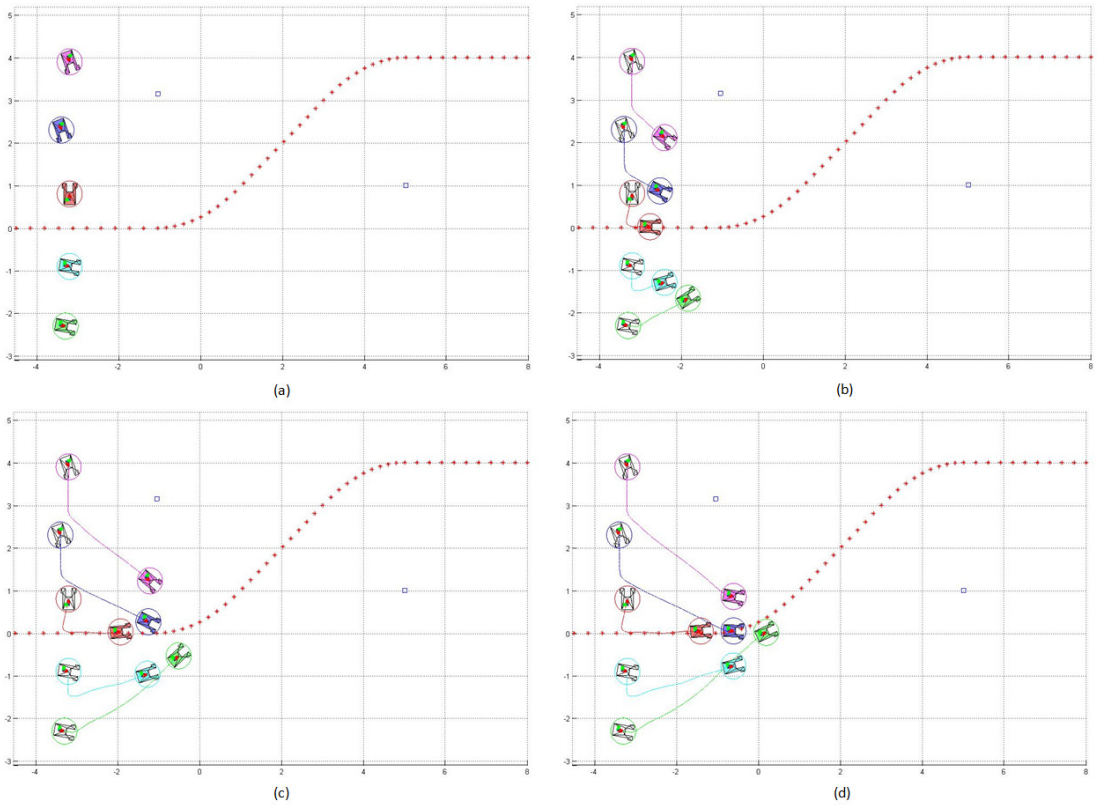


Figure 9.2: Simulation One, First Stage: Transient state of the framework.

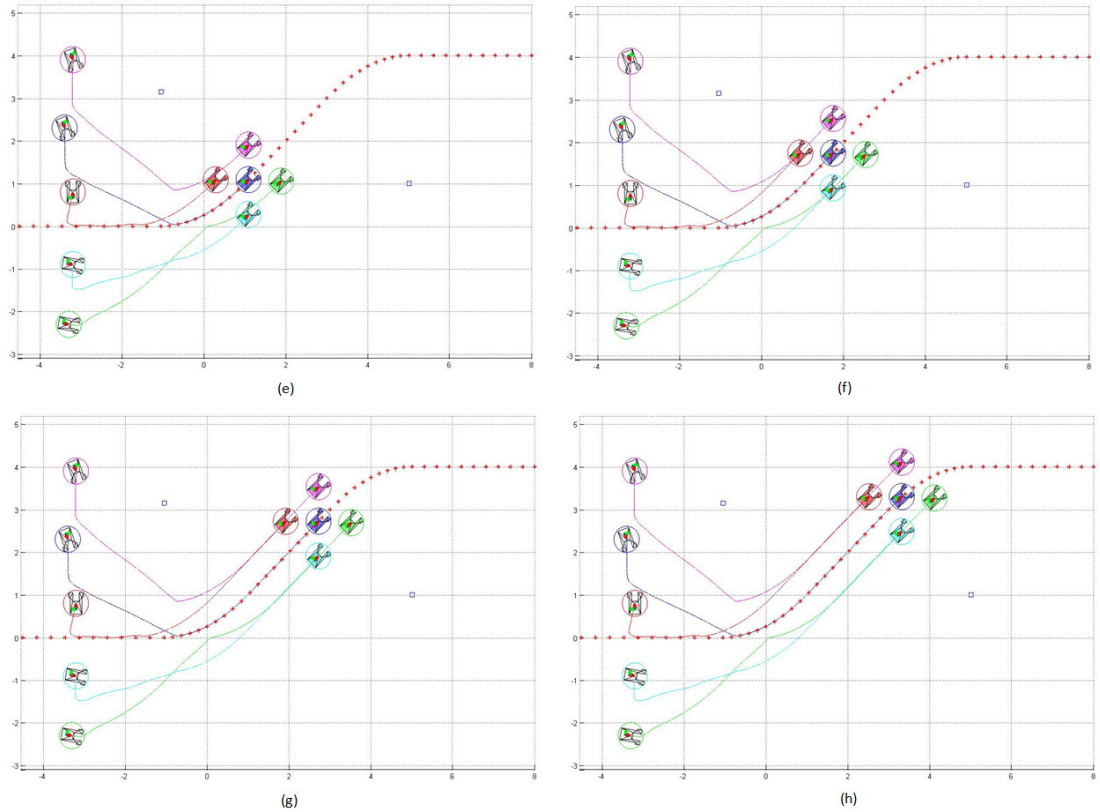


Figure 9.3: Simulation One, Second Stage: Robots keep the formation while moving over the designed route.

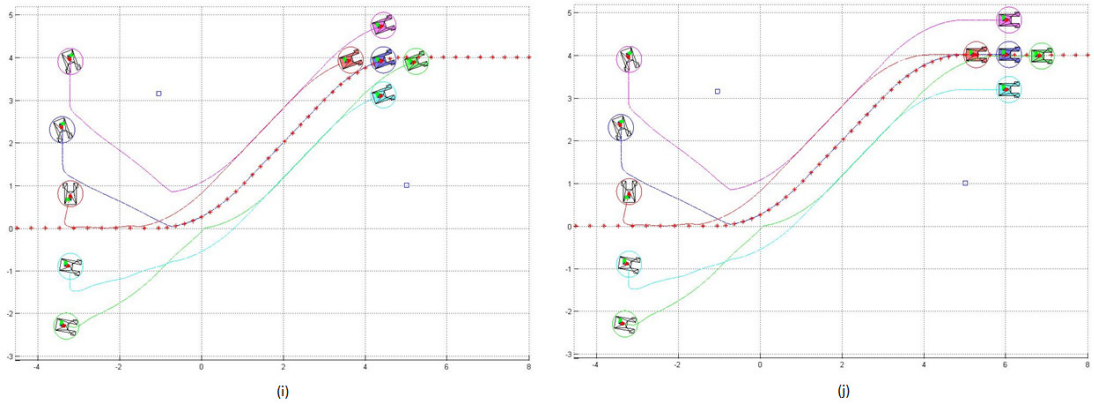


Figure 9.4: Simulation One, Final Stage: The robots have traversed the goal route.

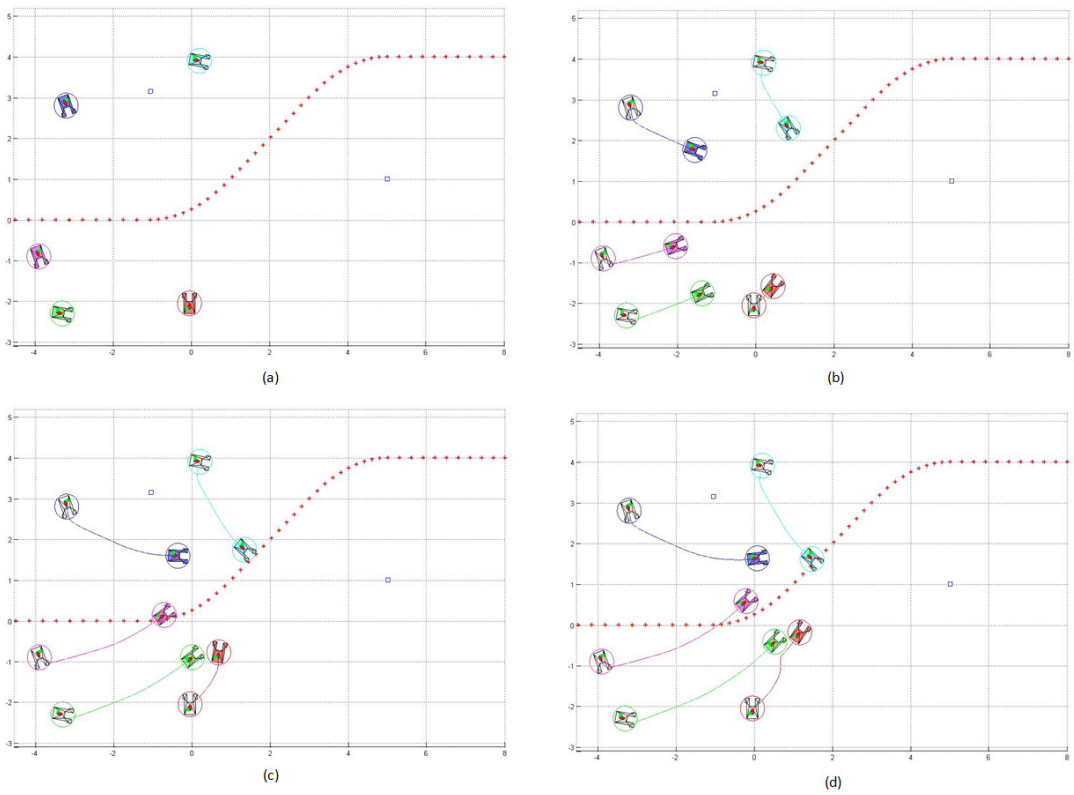


Figure 9.5: Simulation Two. Robots start scattered all around the workspace and try to unite while moving on the direction of the path to follow.

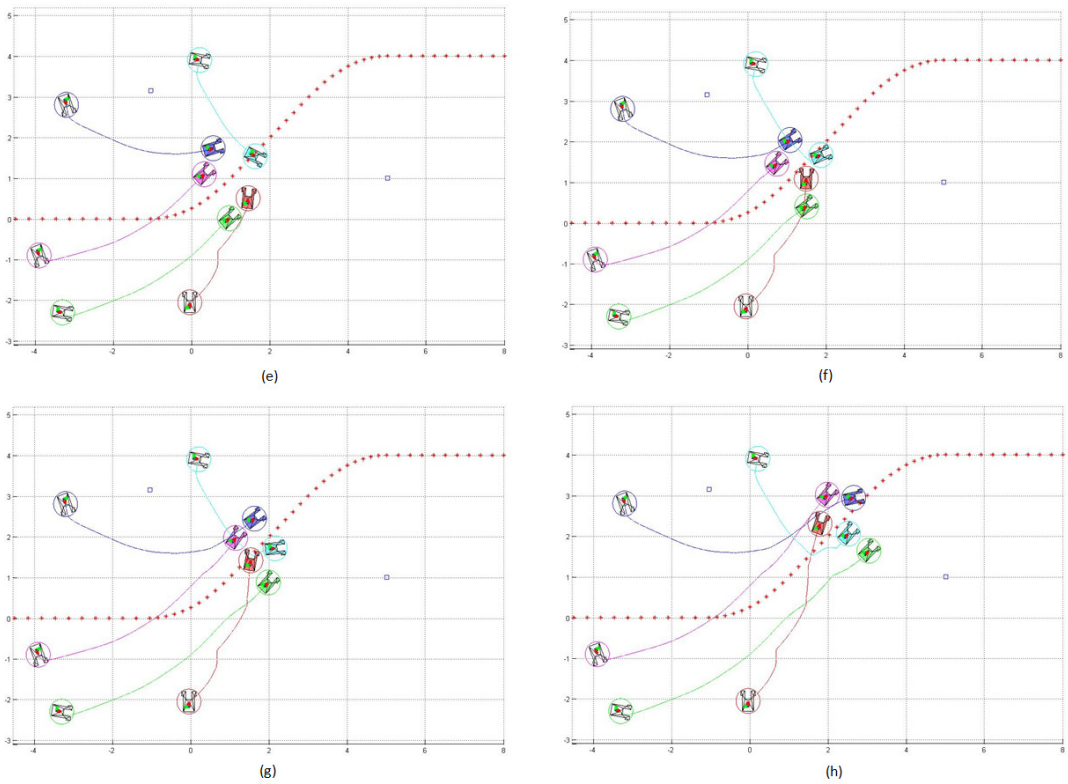


Figure 9.6: Simulation Two. Robots reaccomodate while doing path following to reach their assigned place on the formation. Notice that none of them is colliding.

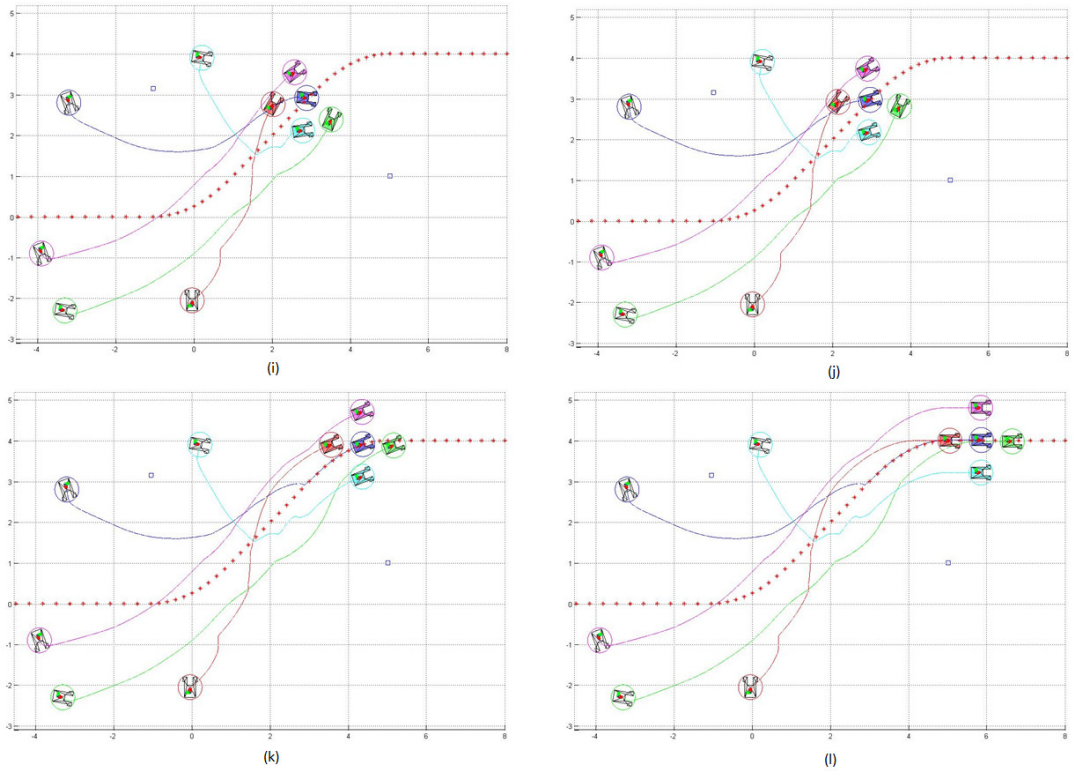


Figure 9.7: Simulation Two. Robots finally achieve formation and continue moving on the path. They have achieve flocking.

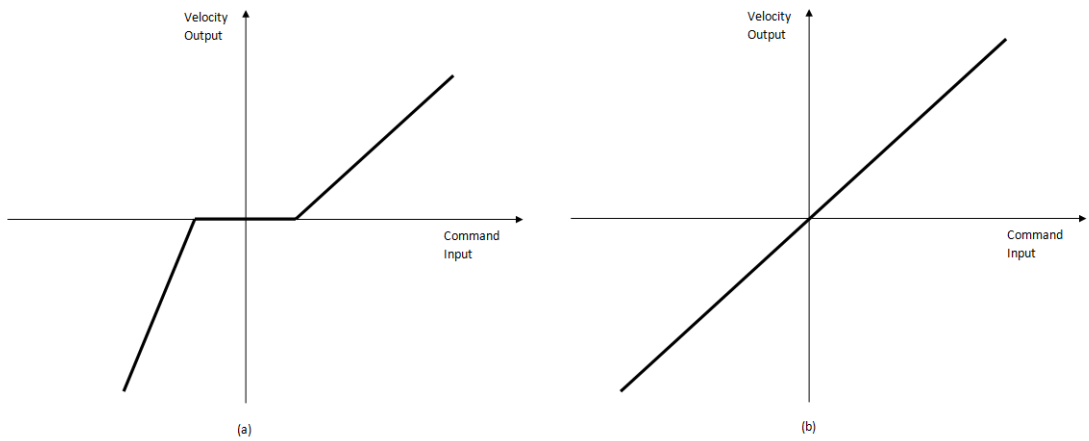


Figure 9.8: Dead band and assumption for the motors. (a) dead band. Note also that the slopes in the regions of actuation are not the same. (b) With some software fixing we can get a behavior close to this one.

battery and the dead band of the robots' motors. Dead band [16] refers to the lack of sensivity or actuation in a particular range of the input; in the case of the robots' motor, the dead band is present around the origin (Fig. 9.8). In practice, the robot does not move for motor inputs below 0.1 in the forward direction and -0.09 in the backward direction. It was also observed that for the function describing the deadband behavior, the slopes are different for the forward and backwards direction. A similar behavior is observed for the angular velocity. However, the experience gained by working with this platforms has enabled us to deduce a correction mechanism that alleviates the effect of dead band and a behavior close to the one displayed on Fig. 9.8(b).

Our control laws require feedback based on the position of the robots. Our localization system, treated with detail in Chapter 8, reports data at a rate of 100 Hz. The experiments' data contains the position and orientation of each robot on a text file and we represent each point the robots visited with a colored frame that shows graphically the position and orientation of the robots local frame, axes $\hat{x}_r$ and $\hat{y}_r$ are represented by the red and green solid lines, respectively; As mentioned before, we have space constrains so the designed path passes over the same space more than once so we represent the path segment in operation with blue marks. For the final snapshot

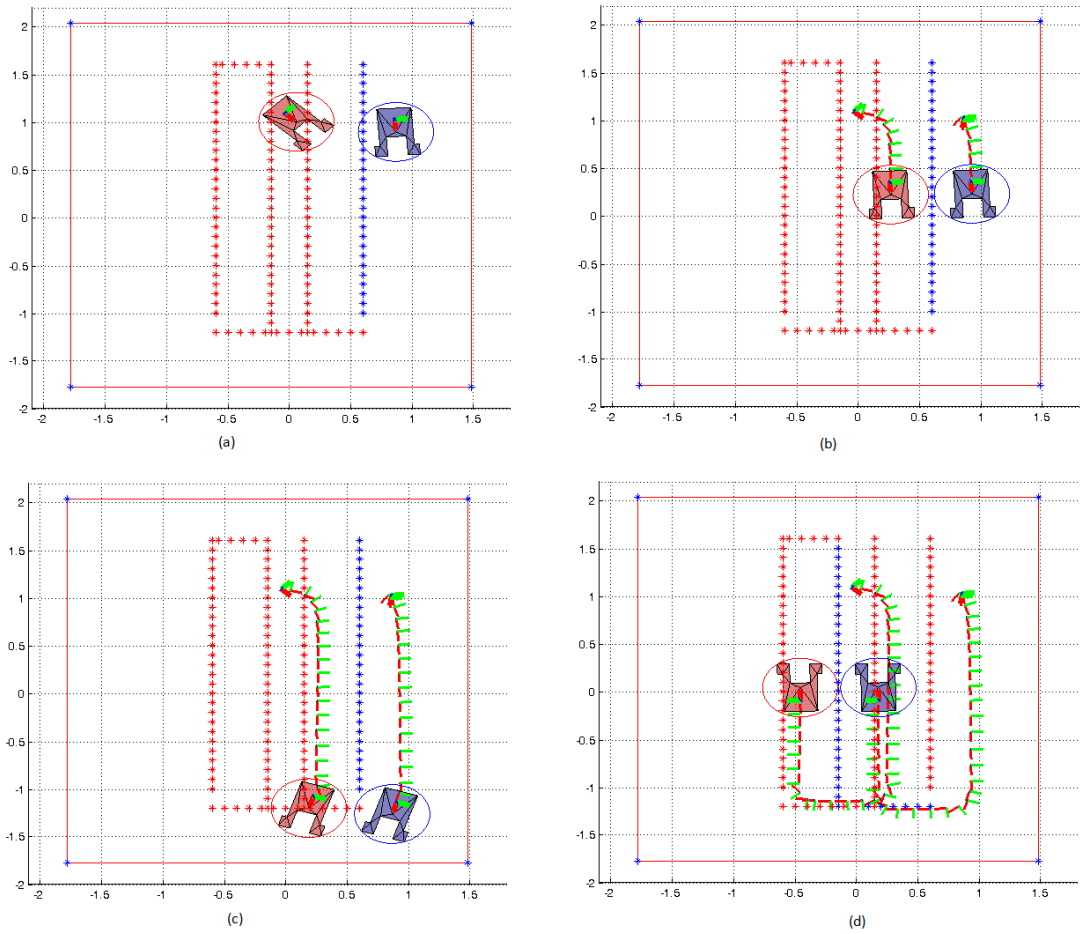


Figure 9.9: Experiment One. Robots move to create a horizontal straight line to do area search on the floor of our laboratory.

of both experiments, we only keep the last frames that represent the robots' position; this is because we want to avoid ambiguity over the traversed path of the robots.

Figures 9.9 and 9.10 show the performance of the two robots during the first experiment. Note that the blue robot moves slower to wait for the red robot to reach the formation and then both of them continue the scanning of the ground keeping a horizontal straigth line. Note that the blue robot is slightly further ahead from the red robot on snapshot Fig 9.10(e), this is because we need to completely stop the robots to take the the picture of the ground without generating blurred images due to the movement. We also include a picture of the experiment, see Fig. 9.11.

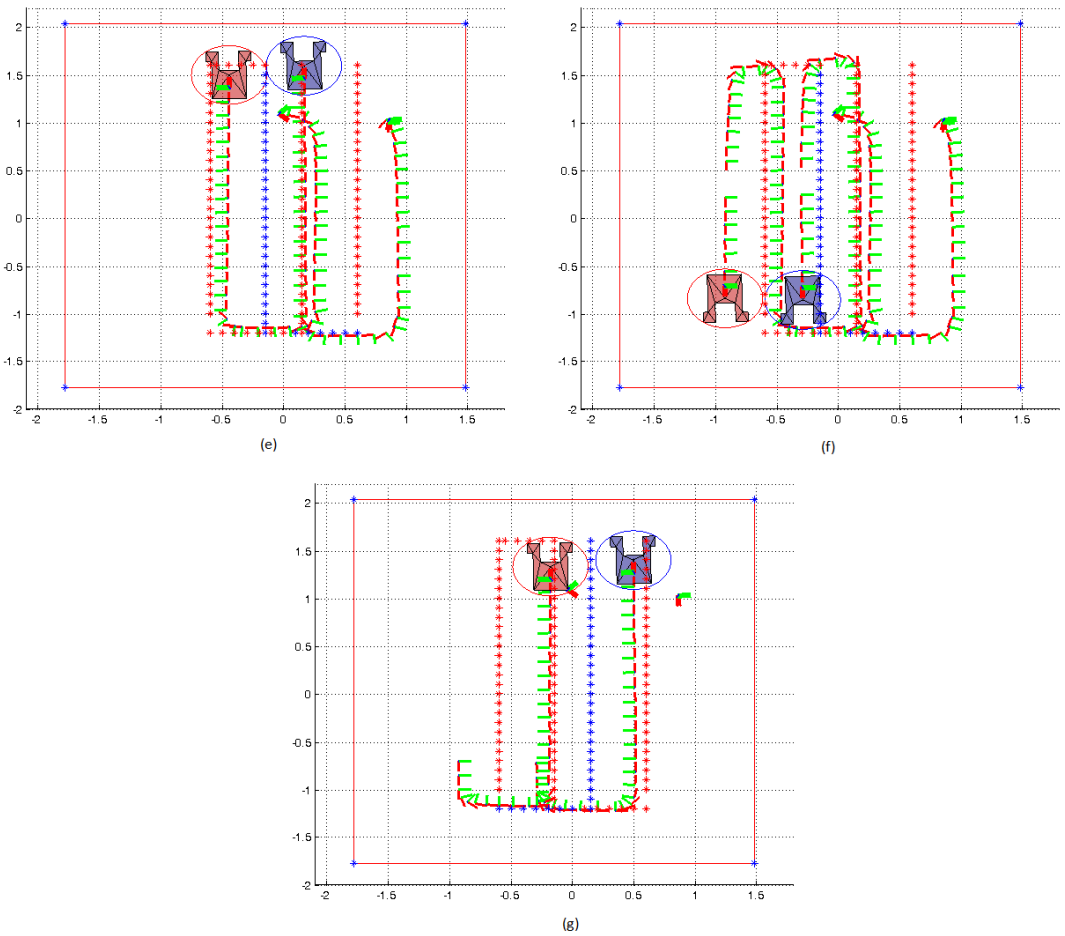


Figure 9.10: Experiment One. Final stage of the experiment

Figure 9.11: Picture of the real experiment one. This picture corresponds to some time between snapshots (c) and (d).

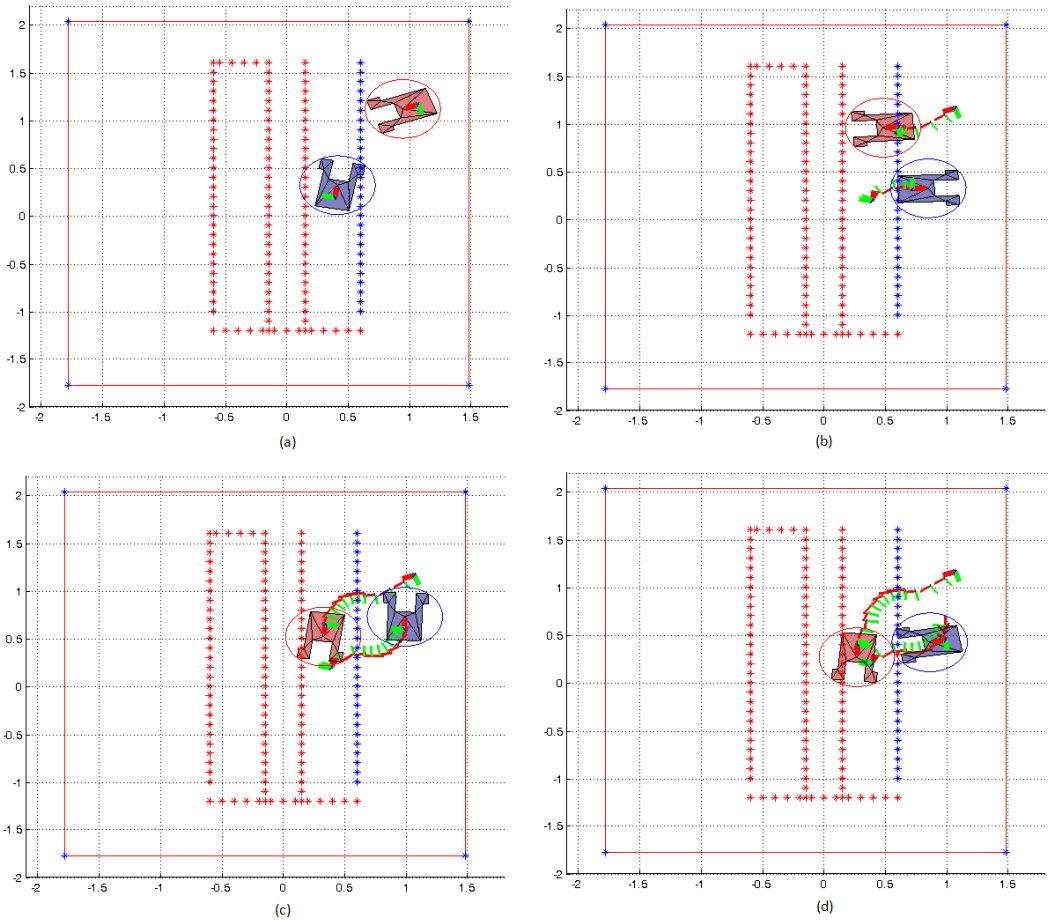


Figure 9.12: Experiment Two. Transient Stage: As robots start in opposite sides to the assigned positions in the formation, it takes them more time to achieve the formation.

Our second experiment is documented of Fig. 9.12, 9.13 and 9.14. This is a more challenging experiment because the robots start at locations that are opposite to the positions assigned on the formation.

Finally, we include the data obtained by the formation deployed on the experiment in pictorial form, see Fig. 9.15. This image was stitched based on the position reported by VICON motion capture system for each robot. The processing and analysis of this picture is beyond the scope of this thesis.

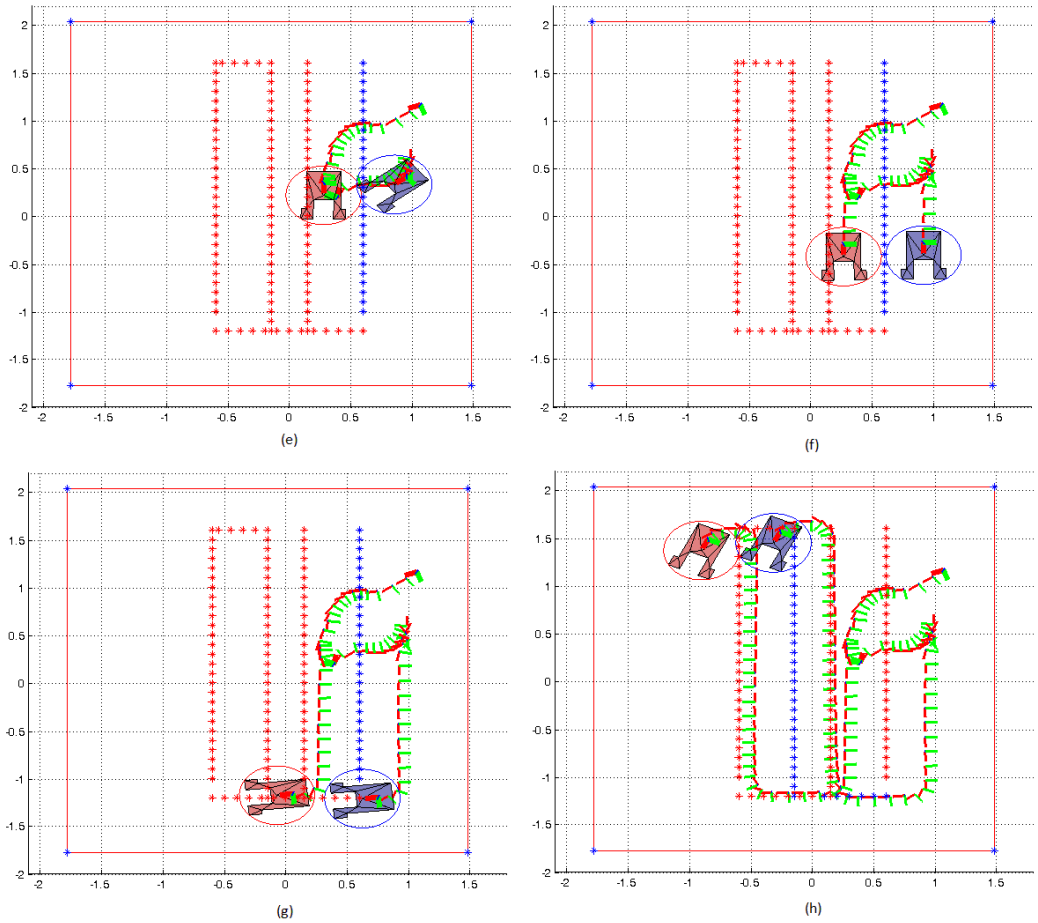


Figure 9.13: Experiment Two. Once they have achieved the formation, they continue doing path following.

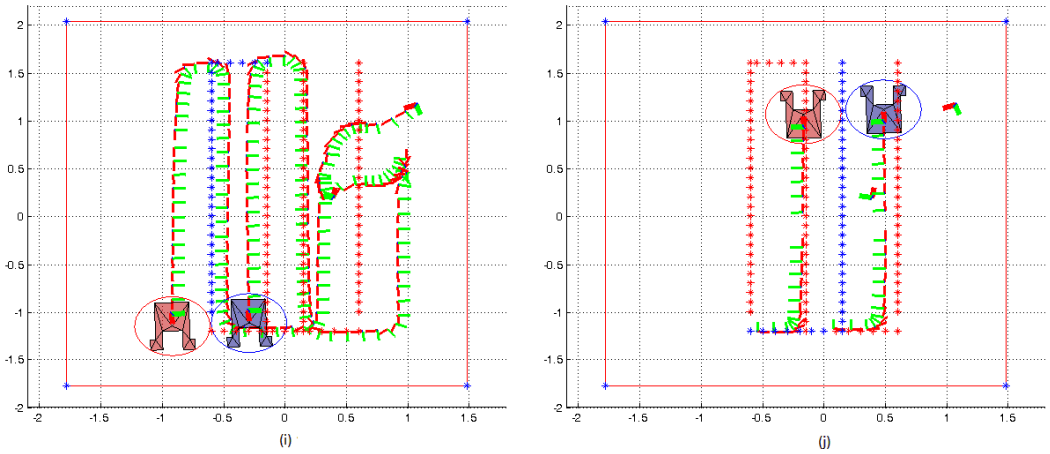


Figure 9.14: Experiment Two. Final Stage: The robots have achieved steady state and are doing flocking until the end.

9.2.1 Implementation Issues

We observed that the fastest robot was exhibiting some chattering motion for a brief period of time, because there is a small difference in the real velocities of the robots (due to the charge state of the battery, the dead band of each motor, slippy floor, etc.). As the fastest robot advances forward, it deforms the formation so the formation control enters into play and forces it to compensate by going back, but as the other robot is advancing and doing its part to compensate for the error too, the fastest robots bounces back to the original path. This was corrected by normalizing the robots velocity and varying the linear velocity's gain according to each robot capability.

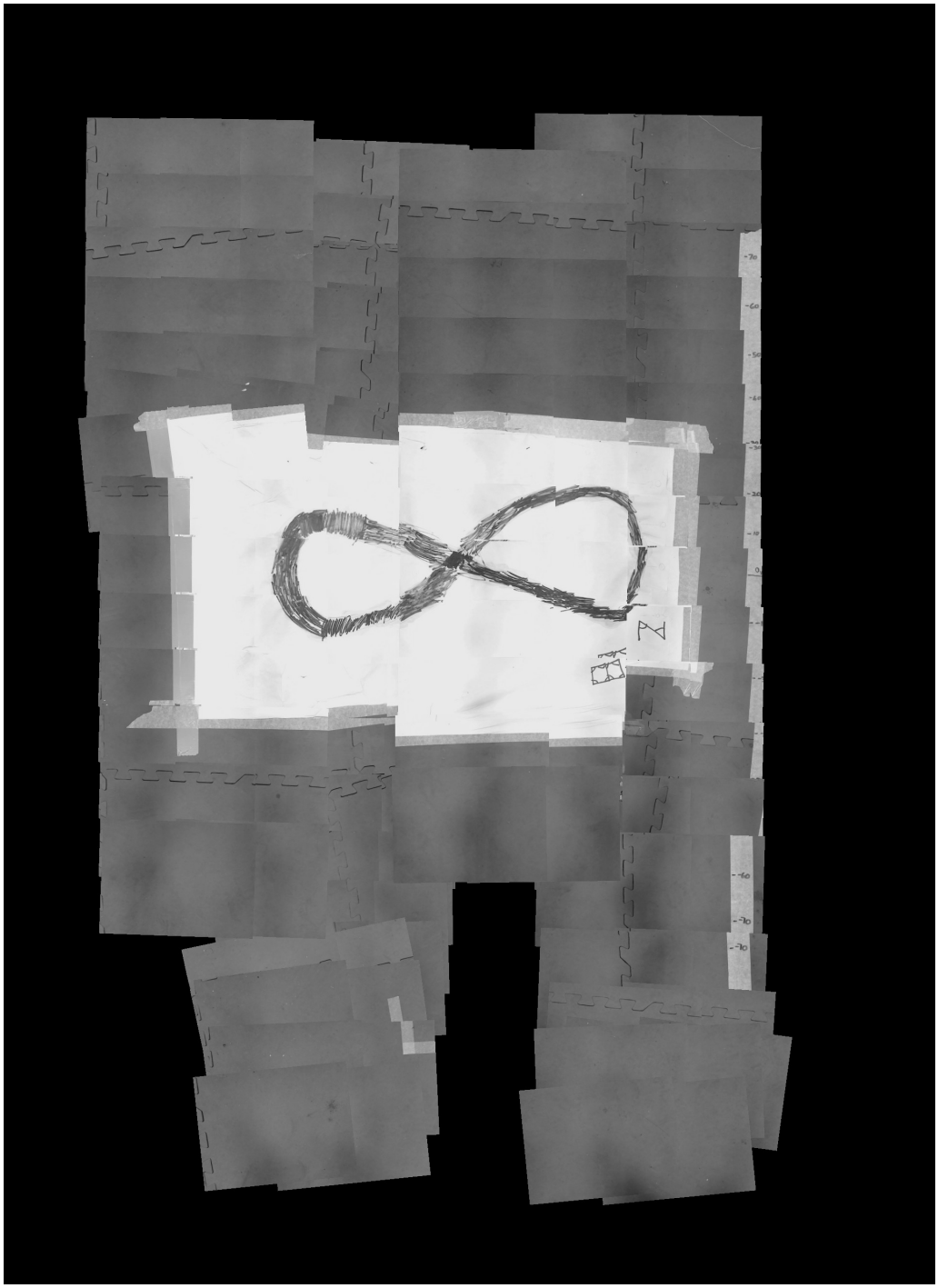


Figure 9.15: Resulting collected data. Courtesy of Prasanna Kannappan who performed the stitching.

Chapter 10

DISCUSSION AND OUTLOOK

10.1 Goals Achieved

Our objective is to create a flexible mobile robotic framework capable of realizing cooperative behavior to increase the sensing capabilities of a group of robots equipped with some particular sensor. This methodology gives rise to a mobile robot group which is autonomous enough to arrange itself into a convenient formation pattern, and then move together following a prescribed path to cover an area of interest.

Our approach is to split the general problem into subproblems. We investigate the state of the art on each sub-topic, identify limitations, and overcoming them with new theory and algorithm. The specific subproblems identified are:

- Formation control: We adopt an existent formation control algorithm based on artificial potential fields and extend it to account for a double integrator dynamics. We additionaly formulate a flocking control term and perform a mathematical analysis that revealed the convergent properties of our modifications despite the appereance of a nonsmooth term in the formulation of the potential field for the formation control. We also identify necessary conditions concerning the attributes of the formation graph and the selection of the incidence matrix for the correct execution of the formation control stage.

- Path following: Existing path following approaches based on artificial potential fields require some sort of switching mechanism to generate path following. We devise a smooth path following algorithm that avoids switching. To do so, we combine vector fields generated for geometrical primitives such as straight and Bézier lines with windowing functions; as a result we get a route composed by several path segments that are added up without switching.

- Synergy of path following with formation and flocking: We were in need of a decoupling mechanism that allowed the accomplishment of the previous controllers at the same time. We realized that the incidence matrix of the (minimum spanning tree of the) formation graph used to get the states for the formation control

routine, provides us with the tool to do such a decoupling. This is because the mapping it performs consist of differences of position, velocity and acceleration among the robots, and eliminates any common bias value within the robots. If we feed all the robots with the same path following control input, this common control component is eliminated through incidence matrix and does not affect the relative position dynamics on which the formation control applies.

10.2 Issues and Limitations

10.2.1 Computational Complexity

There are several mathematical calculations regarding the path following routine that take considerable processor time. However, the windowing functions also provide a practical means to reduce computational overhead. If the calculation of the windowing functions for a particular path segment object is sufficiently small the particular vector field component can be ignored.

The incidence matrix (decoupling mechanism) depends on the chosen pair of robots that take part in the minimum spanning tree, therefore is variable and needs to be calculated when the distribution of pair changes. The incidence matrix is not invertible and despite the fact there is a formula to obtain a generalized inverse, the latter can lead to inaccurate values; hence, the mapping of the formation and flocking control input would be affected. The other option to calculate the generalized inverse is to use decompositions methods but this takes time. We address this difficulty by considering all the $n(n-1)/2$ possible spanning trees, calculate the generalized inverse and store the value on the client program; then when executing the global control, the program just picks the correct generalized inverse. Thus this part of the calculation is performed off-line.

10.2.2 Implementation Issues

We have identified several factors that affect the performance of the whole system. By performance, we mean accuracy in terms of formation stabilization and path following.

10.2.2.1 Hardware Implementation

- Calculation-Estimation of the relative velocities: We implemented a very crude estimator for the measument of the relative velocities based on the data provided by VICON motion capture system. Estimated velocites are corrupted with high frequency noise that mainfests itself in the formation control through prominent oscillation on the movement of the two robots. Improvements on this front can be expected if one uses a well tunned extended Kalman filter.

- Position measurement accuracy: If the specified formation is required to be spatially dense and there is a relatively big uncertainty on the measurements, there will be a high probability of collision.

- Similarity of the robots involved in the formation: If for the same digital input provided by the client program makes two robots move with slightly different velocities, then formation control accuracy is compromised. This issue was addressed here with appropiate tunning of control gains.

- Assignment of tasks to the robots processors: Forcing the robots to take pictures while in motion brings up task scheduling issues: the picture taking process takes about 400 ms and during this time the processor is occupied; during that same time window, motion correction maneuvers may need to be performed and by the time the processor is released it may be too late to avoid collision. This must be addressed with a multi-thread approach.

10.2.2.2 Control Design

- Accuracy in path following: The accuracy in path following can be finely tunned by modifying the values of the constants involved in the formulation of the path segments. In practice, however, it is affected by the accuracy of the VICON motion capture system.

- Accuracy in robot formation: The process of collectin images from ground targets requires the robots to cover the working area in a way that the field of view of each camera overlaps the one of the camera next to it. This places a constraint on how far apart the robots have to be with respect to each other. In our application the formation specification placed the robots fairly close to collision configurations, putting some pressure on the necessary conditions for potential field correctness [42]. During the experiments there were no collisions, yet the pictures taken by the robots were good enough to guarantee overlapping.

10.2.3 Technological Bottlenecks

One important aspect is the design of the physical robot platforms. Despite the fact the robots used in the experiments were manufactured by the same company, they

are not physically identical. They are typically fabricated on a "build on request" policy which implies that the manufacturers do not necessarily use the same components; hence, there are important dimensional and performance variations between any two robots which affect any controller implemented on the robotic platforms.

Parallelization of mathematical routines is another issue. Mathematical operation such as matrix inversion, generalized matrix inversion, eigenvalue calculation, numerical methods and minimum squares techniques are very important mathematical operations that generally have to be employed when manipulating big amount of data. There are some parallelization routines which are proven to be computationally correct but their execution is cumbersome when put into practice.

BIBLIOGRAPHY

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

[2] R.W. Beard, J. Lawton, and F.Y. Hadaegh. A feedback architecture for formation control. In *Proceedings of the American Control Conference*, volume 6, pages 4087 –4091 vol.6, 2000.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Alan S. Brown. The drone warriors, January 2010. Mechanical Engineering.

[5] C. Canudas de Wit and O.J. Sørdalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 37(11):1791 –1797, 1992.

[6] Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990.

[7] Samuel Coogan and Murat Arcak. Formation control with size scaling using relative displacement feedback. *American Control Conference*, 2012.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill, 2001.

[9] Jorge Cortes. Discontinuous dynamical systems: A tutorial on solutions, nonsmooth analysis, and stability, March 2008.

[10] Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52:852–862, 2007.

[11] Felipe Cucker and Steve Smale. On the mathematics of emergence. *Japanesse Journal of Mathematics*, 2:197–227, 2007.

[12] M.C. De Gennaro and A. Jadbabaie. Formation control for a cooperative multiagent system using decentralized navigation functions. In *American Control Conference*, pages 1346 – 1351, 2006.

[13] Carl L. DeVito. *Functional Analysis*. Academic Press INC, 1978.

[14] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. A feedback stabilization and collision avoidance scheme for multiple independent nonholonomic non-point agents. In *Proceedings of IEEE International Symposium on Intelligent Control*, 2005.

[15] 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 the IEEE International Conference on Robotics and Automation*, pages 2430 –2435, 2006.

[16] Jacob Fraden. *HandBook of Modern Sensors*. Springer, 2004.

[17] Yoshihiro Kanno. *Nonsmooth Mechanics and Convex Optimization*. CRC Press, 2011.

[18] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[19] Ji-Wook Kwon and Dongkyoung Chwa. Hierarchical formation control based on a vector field method for wheeled mobile robots. To appear.

[20] S.G. Loizou and K.J. Kyriakopoulos. Navigation of multiple kinematically constrained robots. *IEEE Transactions on Robotics*, 24(1):221 –231, 2008.

[21] Samuel Martin, Arastoo Fazeli, Ali Jadbabaie, and Antoine Girard. Multi-agent flocking with random communication radius. *American Control Conference*, 2012.

[22] D.R. Nelson, D.B. Barber, T.W. McLain, and R.W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519 –529, 2007.

[23] The Stability of Dynamical Systems. *J. P. La Salle*. Society for Industrial and Applied Mathemtics, 1976.

[24] P. Ögren and N.E. Leonard. Obstacle avoidance in formation. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 2492 – 2497 vol.2, 2003.

[25] Petter Ögren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49:1292–1302, 2004.

[26] Petter Ögren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49:1292–1302, 2004.

[27] Kwang-Kyo Oh and Hyo-Sung Ahn. Distance-based sequential formation control of mobile agents by using motion primitives. In *IEEE International Symposium on Intelligent Control*, pages 1464 –1469, 2010.

[28] O.J. Sørdalen and C. Canudas de Wit. Exponential control law for a mobile robot: extension to path following. *IEEE Transactions on Robotics and Automation*, 9(6):837 –842, 1993.

[29] Reza Olfati-Saber and Richard Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520 – 1533, 2004.

[30] Reza Olfati-saber and Richard M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, 2002.

[31] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buch. *Discrete Time Signal Processing*. Prentice Hall, 1998.

[32] Donato Di Paola, Roberta De Asmundis, Andrea Gasparri, and Alessandro Rizzo. Decentralized topology control for robotic networks with limited field of view sensor. *American Control Conference*, pages 3167 – 3172, 2012.

[33] Hasan Poonawala, Aykut C Satici, Nicholas Gans, and Mark W Spong. Formation control of wheeled robots with vision-based position measurement. *American Control Conference*, pages 3173 – 3178, 2012.

[34] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Conference Proceedings in Computer Graphics*, 21(4):25 – 34, 1987.

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

[36] E. P. Ryan. An integral invariance principle for differential inclusions with applications in adaptive control. *SIAM Journal in Control Optimization*, 36:960–980, 1998.

[37] Nilanjan Sarkar, Xiaoping Yun, and Vijay Kumar. Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. *International Journal of Robotic Research*, 13(1):55–69, 1994.

[38] Daniel Shevitz and Brad Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on Automatic Control*, pages 1910 – 1914, 1994.

[39] T.H. Summers, Changbin Yu, B.D.O. Anderson, and S. Dasgupta. Formation shape control: Global asymptotic stability of a four-agent formation. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3002 –3007, 2009.

[40] T.H. Summers, Changbin Yu, S. Dasgupta, and B.D.O. Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *IEEE Transactions on Automatic Control*, 56(12):2778 –2792, 2011.

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

[42] Herbert Tanner and Adithya Boddu. Multi-agent navigation functions revisited. To appear.

[43] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Stable Flocking of Mobile Agents, Part II: Dynamic Topology. In *IEEE Conference on Decision and Control*, pages 2016–2021, 2003.

[44] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, Part I: fixed topology. In *42nd IEEE Conference on Decision and Control*, volume 2, pages 2010 – 2015 Vol.2, 2003.

[45] H.G. Tanner and A. Kumar. Towards decentralization of multi-robot navigation functions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4132 – 4137, 2005.

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

[47] V. Trianni. *Evolutionary Swarm Robotics. Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, volume 108 of *Studies in Computational Intelligence*. Springer Verlag, Berlin, Germany, 2008.

[48] Lung-Wen Tsai. *Robot Analysis*. John Wiley & Sons,Inc, 1998.

[49] John Wilson. *Sensor Technology HandBook*. Elsevier, 2005.

[50] Chika Yoshioka and Toru Namerikawa. Formation control of nonholonomic multi-vehicle systems based on virtual structure. In *Proceedings of the 17th World Congress of The International Federation of Automatic Control*, pages 5149– 5154, 2008.

[51] Michael M. Zavlanos and Kostas J. Kyriakopoulos. Decentralized motion control of multiple mobile agents. In *11th Mediterranean Conference on Control and Automation*, 2003.

[52] M.M. Zavlanos, H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Hybrid control for connectivity preserving flocking. *IEEE Transactions on Automatic Control*, 54(12):2869 –2875, 2009.

[53] Tao Zhang, Xiaqin Li, Yi Zhu, Song Chen, Yu Cheng, and Jingyan Song. Formation and obstacle avoidance in the unknown environment of multi-robot system. In *IEEE International Conference on Robotics and Biomimetics*, pages 729 –734, 2009.

Appendix A

AUTHORIZATION LETTERS

This appendix includes the authorization letters for Fig. 1.1 and Fig. 9.15 .The image in Fig. 1.1 was taken by Shridhar Shah and Fig. 9.15 is a mosaic of a set of images from an experiment conducted by Prasanna Kannappan and Luis Valbuena Reyes.

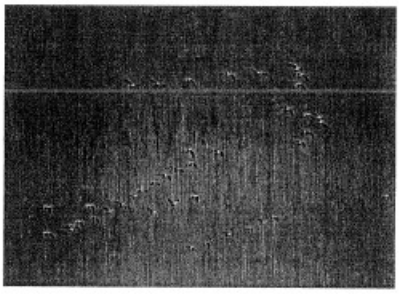
The authorization letter from Shridhar Shah is in Fig. A.1 and the authorization letter from Prasanna Kannappan is in Fig. A.2 .

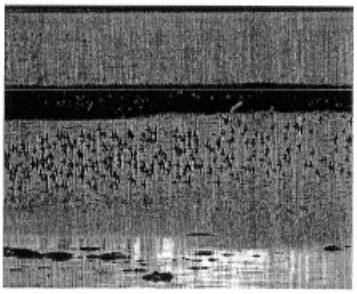
University of Delaware
Newark, DE
19716

November 13, 2012

To whom it may concern:

I, Shridhar Shah, authorize Luis Ariel Valbuena Reyes to use the following to images for his master thesis document titled as: "Flocking with formation control in mobile sensor networks for area search".





Sincerely yours,

Shridhar Shah,

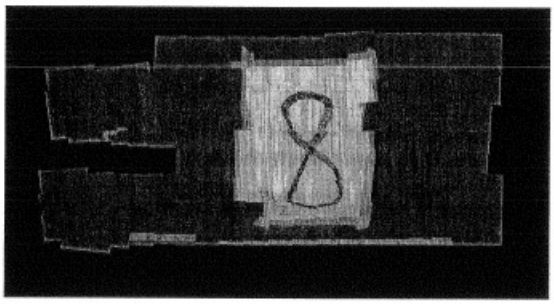
Figure A.1: Authorization Letter One.

University of Delaware
Newark, DE
19716

November 13, 2012

To whom it may concern:

I, Prasanna Kannappan, authorize Luis Ariel Valbuena Reyes to use the following to images for his master thesis document titled as: "Flocking with formation control in mobile sensor networks for area search".



Sincerely yours,

K. Prasanna
10/14/2012

Prasanna Kannappan,

Figure A.2: Authorization Letter Two.