Vers une architecture de commande pour des robots mobiles coopérants non holonomes

Arturo Gil-Pinto

Université Montpellier 2
Sciences et techniques du Languedoc

Thèse
pour obtenir le grade de

Docteur de l'Université Montpellier 2

Formation doctorale: Systèmes Automatiques et Microélectroniques
Ecole doctorale: Information Structures et Systèmes

par

Arturo GIL PINTO

Vers une architecture de commande pour des robots mobiles coopérants non holonomes

Towards a Control Architecture for Cooperative Nonholonomic Mobile Robots

Soutenue le 26 novembre 2007

JURY

Thierry VAL,	Professeur, Université Toulouse 2	Président
Wilfrid PERRUQUETTI,	Professeur, Ecole Centrale de Lille	Rapporteur
Philippe SOUÈRES,	Chargé de Recherches, CNRS LAAS	Rapporteur
René ZAPATA,	Professeur, Université Montpellier 2	Examinateur
David ANDREU,	Maître de Conférences, Université Montpellier 2	Examinateur
Phillipe FRAISSE,	Professeur, Université Montpellier 2	Directeur de Thèse

Université Montpellier 2
Sciences et techniques du Languedoc

Thèse

pour obtenir le grade de

Docteur de l'Université Montpellier 2

Formation doctorale: Systèmes Automatiques et Microélectroniques
Ecole doctorale: Information Structures et Systèmes

par

Arturo GIL PINTO

Vers une architecture de commande pour des robots mobiles coopérants non holonomes

Towards a Control Architecture for Cooperative Nonholonomic Mobile Robots

Soutenue le 26 novembre 2007

JURY

Thierry VAL,	Professeur, Université Toulouse 2	Président
Wilfrid PERRUQUETTI,	Professeur, Ecole Centrale de Lille	Rapporteur
Philippe SOUÈRES,	Chargé de Recherches, CNRS LAAS	Rapporteur
René ZAPATA,	Professeur, Université Montpellier 2	Examinateur
David ANDREU,	Maître de Conférences, Université Montpellier 2	Examinateur
Phillipe FRAISSE,	Professeur, Université Montpellier 2	Directeur de Thèse

à mes parents.

REMERCIEMENTS

Je souhaiterais faire part de ma reconnaissance, de façon générale, au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) ainsi qu'à M. Michel Robert, directeur du laboratoire, de m'avoir accueilli au sein du laboratoire, à l'Université Central du Venezuela (UCV), spécialement au Conseil de Développement Scientifique et Humanistique (CDCH), à la Fondation Gran Mariscal de Ayacucho (FUNDAYACUCHO), pour leur financement.

Je tiens tout particulièrement remercier à mon directeur de thèse, M. Philippe Fraisse ainsi qu'à M. René Zapata pour leurs conseils scientifiques très précieux et la confiance qu'il m'ont accordé tout au long de mon travail de thèse. Je vous remercie de m'avoir donné l'opportunité de m'exprimer sur un sujet aussi intéressant qu'enrichissant ainsi que pour la liberté d'action et la confiance qu'ils m'ont concédées.

Je remercie sincèrement M. Wilfrid Perruquetti, professeur de l'École Central de Lille, et M. Philippe Suères, chargé de recherche du Laboratoire d'Architecture et d'Analyse des Systèmes (LAAS) Toulouse, qui m'ont fait l'honneur d'être les rapporteurs de mon travail, à M. Thierry Val, professeur à l'Université Toulouse 2 qui a présidé le jury, ainsi que M. David Andreu maître de conférences de l'Université Montpellier 2. Je les remercie pour leur regard critique et pertinent, et leurs conseils.

Les mesures et expérimentations ont été possibles grâce aux travaux des stagiaires Guillaume Sauvage, Cédric Noukpo et Angela Dieguez, ainsi qu'à la collaboration et support du permanent M. Oliver Tempier.

Un merci chaleureux au peloton de tous mes amis thésards et ex-thésards, merci pour votre amitié et bonne humeur que je n'oublierais jamais:

Andreea Ancuţa, Carla Aguiar, Jose Marconi, Jean-Mathias Spiewak, David Corbel, Vincent Bonnet, Rogerio Richa, Aurelien Noce, Walid Zarrad, Antonio Bo. C'était très agréable rouler dans un peloton si fort, même dans les étapes travail au labo, comment les étapes de soirées poker, ou les étapes de sports, ou de soirées cinéma, ou de pubs (il a manqué l'étape reine... *rappel toi churrascaïa*). Je suis sur q'on va se rencontrer bientôt de nouveau dans la route, soit en France, soit au Brésil, soit en Roumanie, soit en Tunisie, soit au Venezuela; ou soit dans le Tour de Saint-Laurent d'Aiguze 2008. *Tous pour un et un pour tous!.*

Je remercie egalement à mes collegues, Milan Djilas, Mourad Benoussad, Sebastien Lengagne, Florence Jacquey, Peter Meuel, Michel Dominici, Yousra Ben Zaida, Oliver Parodi et Ashvin Sobhee, merci à vous tous.

Finalement je tiens à remercier à ma famille, très particulièrement à mon frère Yvan, à sa femme Carolina et à ses enfants Eduardo et Ana-Sofía, pour m'avoir acuielli dans leur maison ici à Montpellier. En gros merci à mes parents Freddy et

Nena pour le support et encouragement pour la réalisation de cette thèse.

ABSTRACT

This work proposes a control architecture for a group of nonholonomic robotic vehicles. We present a decentralized control strategy that permits each vehicle to autonomously compute an optimal trajectory by using only locally generated information. We propose a method to incorporate reactive terms in the path planning process which adapt the trajectory of each robot, thus avoiding obstacles and maintaining communication links while it reaches the desired positions in the robot formation. We provide the proof of the reachability of the trajectory generation between the current and desired position of each follower. Simulation results validate and highlight the efficiency and relevance of this method. An integration of the wireless network signal strength data with the vehicle sensors information by means of a Kalman filter is proposed to estimate the relative position of each vehicle in a robot formation set. Vehicle sensors consist of wheel speed and steering angle, the WiFi data consist of reception signal strength (RSS) and the angle of the maximal RSS with respect to the robot orientation. A nonholonomic nonlinear model vehicle is considered; due to these nonlinearities an Extended Kalman Filter EKF is used. Simulation and experimental results of the proposed estimation strategy are presented.

Keywords: Mobile robot, Cooperative systems, Optimal control, Mobile robot motion-planning, Global positioning system, Wireless LAN.

CONTENTS

NOTATION

GPS	Global Positioning System
RSS	Reception Signal Strength
x, y	Cartesian coordinates
θ	Robot orientation angle
u_1, u_2	Linear and steering velocity control
DVZ	Deformable Virtual Zone
ρ	Satellite pseudo range
EKF	Extended Kalman Filter
DGPS	Differential Global Positioning System
$\mathscr{C}$	Robot set
i, j	Leader and follower index
$\mathbf{q}$	Robot state
$\mathscr{E}$	Formation node
$\mathscr{V}$	Formation vertices
$\mathscr{G}$	Graph
l	Relative distance
γ_{ij}	Relative robots (i, j) angle
ξ	Steering angle
v	Linear velocity
ω	Angular velocity
τ_v, τ_ω	Time constants
R	Robot wheels radius
L	Robot length
β_{ij}	Relative angle $\theta_j - \gamma_{ij}$
$\mathbf{r}_{ij}$	Leader-follower state
J	Optimization performance index
$\mathbb{R}$	Real set
z	Flat output
S, U	State and control trajectory
C	B-spline parameter
B	B-spline basis function
QP	Quadratic Programming
SQP	Sequential Quadratic Programming
ICR	Instantaneous Rotation Center
$\mathbf{R}$	Instantaneous Rotation Radius
$()_{lead}, ()_{foll}$	Leader and Follower variables
r_{ss}	Steady-State instantaneous rotation radius
V	Lyapunov function
T_s	Trajectory generation sampling time
τ	Normalized time
$\mathbf{A}$	System matrix
$\mathbf{B}$	Control matrix
dB	Decibels
GHz	Giga Hertz
$\mathbf{y}$	System output
ZOH	Zero Order Holder discrete approximation

LIST OF FIGURES

1

Introduction

Over the past several centuries, progress in sparing humans from hard labour has accelerated. Throughout the world, animals and machines now perform many work activities, providing many with easier lives, greater safety and more independence. The drive to find substitutes for humans in hazardous environments and fatiguing activities has been one of the principal motivations for the search for autonomous and secure systems. The Egyptian, Roman and Greek civilizations took advantage of the more developed sensors of dogs and domesticated them for use in exploration and hunting. Over time, even more autonomous solutions were found with the introduction of various mechanical innovations. The Greeks used their understanding of hydraulic principles to operate statues that imitated human characteristics. In 12th century China, a south-pointing chariot was equipped with a statue that was turned by a gearing mechanism attached to the chariot wheels in such a way that it continuously pointed south. Hence, using the directional information provided by the statue, the charioteer could steer a straight course. In more recent times, such mechanical innovations have been combined with progress in computing power and theoretical and technological advances, giving rise to the field the robotics. Robotics as it is known today is a vast interdisciplinary science comprising many fields of research, including vision, sensing, dynamics, motion planning and control, locomotion, and design. One of the most basic problems, however, remains the motion planning and control of robots. Robot motion planning deals with finding a feasible trajectory for a robot moving in an environment with obstacles.

Since a single robot, no matter how capable, is spatially limited, the study of multi-robot systems has became a major focus in the robotics research community. Although the field of multiple robot systems naturally extends the research on single robot systems, it is also a discipline in itself: multiple robot systems can accomplish tasks that single robots cannot. Multi-robot systems are also different from other distributed systems because of the implicit real-world environment.

The study of multi-robot systems entails the analysis of cooperation and coordination laws between the robots. Cooperation in multi-robot systems was defined by Farinelli *et al* [1] as the ability of a system to cooperate in order to accomplish a specific task; coordination was defined as the set of mechanisms and interactions used to obtain this cooperation. In chapter 2, we review and extend these concepts.

Coordination in a multi-robot system can be achieved by means of explicit communication between robots via sonar for underwater and terrestrial robot systems and radio communication for aerial systems.

Coordination by means of explicit communication allows the robot formation to be spread over a much wider area compared with other coordination mechanisms like vision. One of the chief concerns in real-world applications of multi-robot systems is security, designed as the ability of the individual robots to continue interacting with the other robots while they perform a predefined team task. To ensure this coordination between robots, the communication also has to be carried out in some autonomous way by the set of robots.

As it will be discussed, the aim of this thesis is to investigate motion planning in nonholonomic multi-robot cooperative systems. We were motivated by applications in unstructured environments, especially for the performance of outdoor tasks like rescue, surveillance and exploring. The multi-robot system is subject to many constraints like, such as obstacles in the environment, many possible paths for each robot, the avoidance of communication losses, and the many ways in which the robots can be organized in the formation. For this reason, the optimal solution might be a highly decentralized path planning procedure for this type of system. In this approach, for example, the communication variables are taken into account to ensure the interaction of the robots while the task of the formation is being accomplished in the unstructured environment.

1.1 Approach

Many approaches to planning and controlling the trajectory of nonholonomic mobile robots have been described in the literature and will be presented in chapter 2. Trajectory planning can be classified as either online or offline. The online methods use knowledge of the current system state and its environment to plan the motion for the next time instant. Since these strategies do not require any precomputation, they are suitable for tasks that do not require any specific constraints to be satisfied along the trajectory, i.e. for tasks whose only requirement is the desired goal configuration. Such schemes are rarely concerned with finding and optimizing a feasible trajectory. On the other hand, the offline approaches rely on prior information about the desired task and the environment configuration, taking into account the systems constraints. Consequently, a feasible open-loop trajectory can be found and optimized.

In multi-robot systems control, the trajectory planning of each nonholonomic robot also depends on the desired cooperation between the robots, i.e. the decision about whether a predefined geometrical pattern is needed between the robots. The robots can be coordinated in a centralized or decentralized way to achieve the desired tasks. In a centralized organization, the trajectory planning relies on information about all the robot positions and there is thus a central trajectory computing agent. In contrast, decentralized coordination uses only local information for each robot. Therefore, no central computing agent is required.

In the following work, a new secure control strategy for multiple car-like vehicle formation is proposed. Distributed trajectory planning is used to obtain an open-loop trajectory for each robot in the formation. The desired trajectory for each robot is a locally optimal solution; in this optimization problem a reactive term is incorporated for the obstacle avoidance constraint in unknown environments. We also incorporate vehicle and trajectory constraints, such as a saturation of the controls and the desired position in the formation for each robot.

In real applications of multi-robot systems, security is a major consideration in the design of the control strategy. For instance, to ensure that the breakdown of a single mobile robot does not jeopardize the mission of the whole formation, the required communication between robots has to be maintained and the data (i.e. position) measurement has to be guaranteed. We propose the use of a global positioning system (GPS) for the position measurement and a WiFi network as the communication platform. For this purpose, the communication link between robots is secured by measuring the WiFi reception signal of each robot and incorporating it into the trajectory generation scheme.

1.2 Problem Definition

The aim of this thesis is to study the control of cooperative terrestrial nonholonomic autonomous vehicles, which can be formulated as follows: Given: the instantaneous position and orientation of a leader agent and the initial positions of a set of n robots in the workspace generate a continuous trajectory for each robot, with each robot avoiding obstacles, keeping a predetermined geometrical pattern with respect to the leader agent, and being subject to the dynamical constraints of vehicles. We elaborate a decentralized trajectory generator that computes a feasible trajectory for each robot in the formation while respecting the dynamical constraints of vehicles and the desired geometrical pattern of the formation. In addition, we prove the stability of this approach through simulations and experiments.

1.3 Contribution

In this thesis, we develop two specific contributions to the control of multi-robot systems, which can be summarized as follows:

- First, we develop a general, highly decentralized strategy to control formations of cooperative nonholonomic robots. Under this strategy, the highlights of online and offline path-planning techniques are summarized in a single control approach of nonholonomic cooperative robots. A feasible trajectory is computed for each robot; if any obstacle is detected, the trajectory is modified using a defined reactive term. Then the stability condition of the strategy for a formation of n robots is also shown. This trajectory generation process is based on an optimization problem for which several optimization criteria can be included, i.e. minimal communication link losses or minimal time trajectories or composites of them.

- Second, we propose an approach using the reception signal strength (RSS) of the wireless communication to secure the communication links between robots. By measuring the reception signal of the wireless communication architecture, we develop a strategy to guarantee these communication links. Also, the use of the RSS is proposed as an alternative relative-positioning mechanism for the robots when the classical positioning systems like GPS are unavailable.

1.4 Thesis Outline

Chapter 2 presents the previous work in mutli-robot system control and nonholonomic path-planning fields. In the first part, the approaches to control multi-robot systems are presented, as well as the online and offline methods for the path planning and control of nonholonomic robots, such as trajectory optimization problems and reactive methods for obstacle avoidance. In chapter 3, we develop a general mathematical model for decentralized control of nonholonomic robot formations, including dynamical constraints in the framework of the calculus of variations to achieve an optimal solution. A numerical method is then proposed to obtain feasible, fast and accurate solutions. Also in this chapter is formulated the stability conditions for the proposed strategy. For a geometrical formation of robots, each robot is subject to different desired orientation velocities, depending on its position in the formation. For example, for the rotation of a delta formation, the farthest vehicle with respect to the instantaneous centre of rotation of the desired formation pattern can be subject to higher velocities than the maximal admissible vehicle velocity. We develop the constraints in manoeuvring the desired formation to keep the robot formation by inserting these constraint equations in the computation of the agent leader's trajectory. Last, we present our study of the transit-time performance of the proposed strategy.

In Chapter 4, a obstacle avoidance approach is proposed. We define a reactive term which is to be included into the trajectory generator. This reactive term is based on a Deformable Virtual Zone (DVZ) [4] that surrounds each vehicle. By computing the virtual zone deformation caused by the obstacles the trajectory generator search a path that minimizes this deformation. Hence the feasible trajectory is adapted for obstacle avoidance.

Chapter 5 includes the estimation of the reception signal level of the wireless communication system into the decentralized trajectory generator in order to maintain the communication link between robots. By using the signal level we can estimate the leader-follower relative position if the global positioning system (i.e GPS) is no available.

In Chapter 6, a feedback controller is proposed for the leader-follower model. The robot is stabilized about the reference path by defining a linear quadratic regulator (LQR). The controlled variables is the relative leader-follower position. Finally in chapter 7, the final remarks and the proposed future works are detailed.

2

State of the Art

In this chapter, we survey the state of the art in nonholonomic multi-robot systems. We first describe and contrast the studies on different multi-robot control strategies. Then, we present the principal approaches for nonholonomic mobile robot control. In particular, we focus on trajectory planning, as well as the stabilization procedures for the car-like robots. Last, advances in the robotics field are highlighted with special attention to positioning and communication systems as applied to robotics.

2.1 Cooperative Multi-Robot Systems

Multi-robot systems are currently a major focus of research in the field of robotics. Multi-robot systems can accomplish tasks that a single robot cannot, since a single robot will ultimately be spatially limited. In 1989, Brooks and Flynn from MIT's Artificial Intelligence Lab proposed what they called a *radical idea* in the solar system exploration field: *replace a large rover by a collection of small rovers* [5]. They said that *the cost per kilogram of the rovers would be greatly reduced from the economy of building multiple copies. Lower reliability for each individual rover would be acceptable, as failure of a single rover would not jeopardize the whole mission. Indeed the mission could be planned with a particular reliability expectation that was below 100%. Upon landing either together or in smaller groups, the rovers would disperse covering wide ranges over the surface.* This radical idea presents the main motivation for the use of multi-agent robotic systems.

Among the first studies on multi-robot systems were those of Fukuda (cellular robotics [6]), and other MIT researchers working in the field of robotic societies of up to twenty mobile robots: [7], [8]. Other pioneering efforts in multi-robot systems study came from Stanford [9], and Carnegie Mellon [10], universities.

More recently, multiple autonomous mobile robots have been proposed for use in rescue missions [11], [12], exploration [13],[14], [15], military applications, and even entertainment with robotic soccer teams [16],[17].

A multi-robot system can be defined as *a set of robots operating in the same work space* [1]. In addition, cooperative behaviour is settled as: *Given some task specified by a designer, a multiple-robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the mechanism of cooperation or coordination), there* *is an increase in the total utility of the system.* [18].

In the following sections, we present the principal features of cooperative multi-robot systems. We discuss the main works in this field, with special attention given to the formation of nonholonomic robots.

2.1.1 Multi-Robot Systems Classification

Given the variety of designs of multi-robot systems, it is useful to have them organized. [19], [18],[1], developed the taxonomy of these systems. Farinelli's [1] classification relies on the coordinative nature of the robot set. This taxonomy is represented by the hierarchical structure shown by Figure 2.1.

At the cooperative level, we can distinguish cooperative systems from non-cooperative ones. A *cooperative system* is composed of robots that operate together to perform some global task. Farinelli's work, also considers the *coordinative level.* Coordination is a cooperation in which the actions performed by each robotic agent takes into account the actions executed by the other robotic agents in such a way that the whole ends up being a coherent and high-performance operation. Organization level introduces a distinction in the forms of coordination, distinguishing *centralized* approaches from *distributed* ones. In particular, a centralized system has an agent (leader) that is in charge of organizing the work of the other agents; the leader is involved in the decision process for the whole team, while the other members can act only according to the leader's directions. In contrast, a distributed system is composed of agents that are completely autonomous in the decision process with respect to each other; there is no leader in such cases. The classification of centralized systems can be further refined depending on the way the leadership of the group is played. Specifically, strong centralization is used to characterize a system in which decisions are taken by the same predefined leader agent throughout the entire mission, while in a weakly centralized system more than one agent is able to take the role of the leader during the mission.

Instead of a classification based on coordination as proposed by Farinelli *et al*; the classification of Cao *et al.* [18] relies on the structure of the robot set. According to this classification, a multi-robot system can be homogeneous or heterogeneous. A group of robots is said to be *homogeneous* if the capabilities of the individual robots are identical, and *heterogeneous* otherwise. Then, the centralized or decentralized

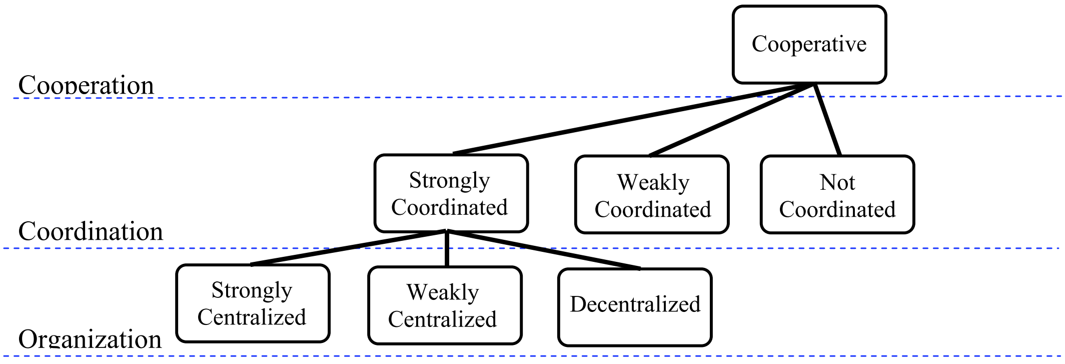


Fig. 2.1: Multi-robot system taxonomy, Farinelli *et al* [1].

architecture is described. Similar to Farinelli's description, Cao *et al.* defined centralized multi-robot systems as formations characterized by a single control agent. A decentralized organization lacks such an agent. They included two types of decentralized organizations: *distributed architectures* in which all agents are equal with respect to control, and *hierarchical architectures* which are locally centralized. In addition, hybrid centralized/decentralized architectures are also described. In these architectures, there is a central planner that exerts high-level control over mostly autonomous agents. A hybrid control architecture for autonomous robot platoons was presented by [20]. In this strategy each robot executes its control locally but relies on the state of all robot set.

One of the main objects of study in multi-robot systems research is the communication or interaction between the robots. [18], [19], studied the different communication structures in these systems. The *communication structure* of a group determines the possible modes of inter-agent interaction. Three main structures are discussed in these works. First is communication or *interaction via environment*: this occurs when the environment itself is the communication medium, and there is no explicit communication between agents. This type of interaction between robots is also known as *stigmergy* and examples can be found in [21]. Another typical structure is the *interaction via sensing*: this refers to local interactions that occur between agents as a result of them sensing one another, but without explicit communication. An example would be vision by means of omnidirectional cameras [22]. Last is *interaction via communications*: this involves explicit communication with other agents, by either directed or broadcast intentional messages. Because architectures that enable this form of communication are similar to those of communication networks, many standard issues from the field of networks arise, including the design of network topologies and communications protocols.

2.2 Nonholonomic Systems

Mobile robots are useful for transport, inspection and intervention in hostile environments. Therefore, the use of mobile robots agents for multi-robot systems applications is becoming increasingly prevalent in academics, military and industrial sectors because their efficiency and flexibility.

The motion of a wheeled mobile robot is subject to nonholonomic constraints due to the rolling constraints of the wheels, which make a motion perpendicular to the wheels impossible. Because of these nonholonomic constraints, the implementation of multi mobile robots is a challenging problem for the multi-agent theory, as well as for the nonlinear control theory.

The analysis and study of multi-robot systems differs from that of multi-agent systems because of the issues arising when dealing with a physical environment, such as uncertainty and the incompleteness of the information acquired from the environment. In fact, the need to cope with the acquisition of knowledge from a real environment makes the experimental evaluation of multi-robot systems much more challenging. In addition, the forms of cooperation used in multi-robot systems need to take into account the uncertainty, limitations, and mistakes arising from the processing of sensor information.

Nonholonomic multi-robot system naturally extend the problems of studying a single nonholonomic autonomous vehicle. Consequently, an examination of the latest research efforts in autonomous multi-agent systems and the control of nonholonomic mobile robots is called for.

2.2.1 Control of Nonholonomic Systems

The word holonomic (or holonomous) is comprised of the Greek words meaning integral and law, and refers to the fact that such constraints, given as constraints on the velocity, may be integrated and reexpressed as constraints on the configuration variables [23]. Examples of holonomic constraints are length constraints for simple pendula and rigidity constraints for rigid body motion. In contrast, nonhonolonomic mechanics describes the motion for systems with nonintegrable constraints; for example, constraints on the system velocities that do not arise from the configurations alone. Classic examples are rolling and skating motions.

The principal characteristics of nonholonomic mechanical systems can be illustrated by the unicycle system example (Fig. 2.2).

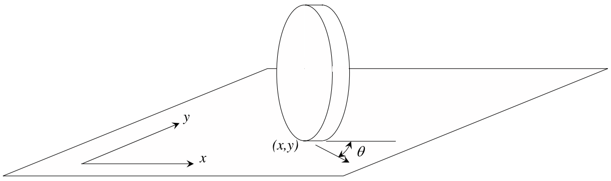


Fig. 2.2: Unicycle system

The kinematic representation of the unicycle system with steering speed $\dot{\theta} = u_1$ and rolling speed $v = u_2$ as controls can be modelled by the equation:

$$\begin{aligned} \dot{x} &= u_2 \cos(\theta) \\ \dot{y} &= u_2 \sin(\theta) \\ \dot{\theta} &= u_1, \end{aligned} \tag{2.1}$$

with the nonholonomic constraint:

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) = 0, \tag{2.2}$$

Therefore, the allowed velocities (u_1, u_2) have to get into the null space of the restriction matrix $a(\mathbf{q})$, with $\mathbf{q} = [x, y, \theta]$:

$$a^T(\mathbf{q}) = [\sin\theta \quad -\cos\theta \quad 0] \Rightarrow \mathscr{N}(a(\mathbf{q})) = span\left\{ \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}. \tag{2.3}$$

The condition (2.3) implies that for each configuration $\mathbf{q}$ there is only two possible movements. The first vector field $g_1 = [\ \cos\theta \quad \sin\theta \quad 0\ ]^T$, represents the vector field of the rolling forward movement of the unicycle at unit velocity, and $g_2 = [\ 0 \quad 0 \quad 1\ ]^T$, its spinning counterclockwise at unit velocity. Hence, the unicycle capacity of rolling forward and backward and spinning in place can be represented by the vectors fields g_1, g_2. The graphical representation of vector fields for the unicycle example can be seen in the Figure (2.3).

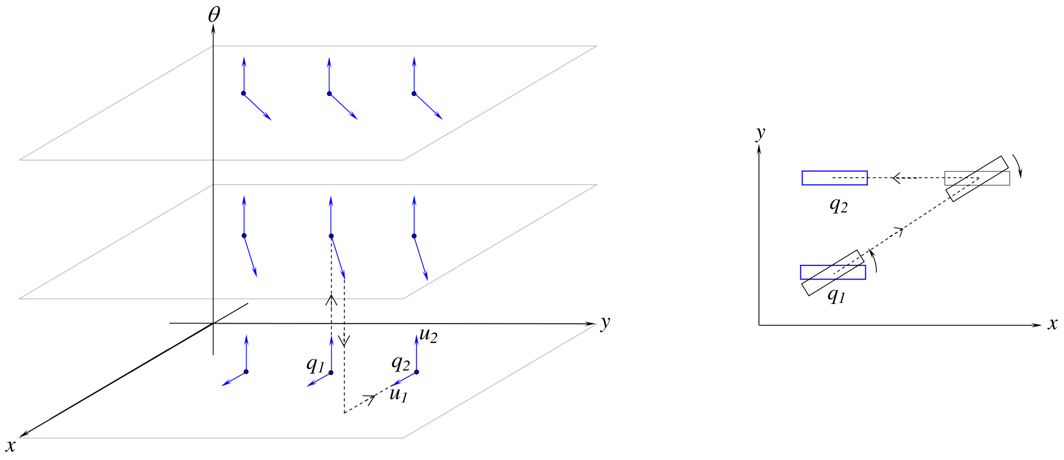


Fig. 2.3: Unicycle's vectorial champ. q_1, q_2 manoeuvring

This restriction over the possible state evolutions in nonholonomic systems has to be taken into account in the design of the control strategy.

In the literature, two main strategies to control wheeled mobile robot control systems have been described: open-loop and closed-loop (feedback).

The open-loop strategies seek feasible trajectories in the free configuration space that connect an initial configuration with a final one. They take into consideration several criteria like collision-avoidance, the shortest path, and minimum control effort. The shortest path with a lower bound on its radius between two oriented points in the plane was studied by Dubins [24]. Such as path is composed of at most three segments of straight lines and arcs. In this case, according to Reeds and Shepp [25], more manoeuvrability is added to the shortest path by using at most five segments of arcs and straight lines. Laumond [26] proposed an algorithm for planning manoeuvres and collision-free paths for circular nonholonomic robots whose turning radius was lower bounded. These open-loop techniques do not compensate for disturbance, system model errors or changing configuration space.

In the closed-loop procedures, the input is a function of the state to compensate for errors and disturbances. Generally, there is no smooth feedback control law that makes a given configuration asymptotically stable. This means that the class of stabilizing controllers should be suitably enlarged so as to include nonsmooth and/or time-varying feedback control laws (i.e [27]). This is a consequence of Brockett's theorem [28], and was also discussed in [29] and has also been discussed in [23]. Given the lack of smooth state feedback laws, works on path following and tracking rely on non-zero reference motion. A survey of local tracking problems for a cart was presented [30]. Ailon *et al.* [31] presented a controllability study of the trajectory tracking problem of a front-wheel drive car-like vehicle. Other control strategies have been used for local tracking control of wheeled vehicles, such as PID controls [32] or sliding mode [33], [34], for example.

By reason of the assumption of non-zero reference motion, stabilization about a fixed configuration cannot be treated by these tracking approaches. Consequently, since there is no smooth control law that stabilizes wheeled mobile robots about a fixed configuration, other classes of feedback strategies have been considered, i.e. discontinuous and time-varying control laws. Nonsmooth controls have been proposed to stabilize nonholonomic vehicles [35],[29], [36],[37].

For open-loop and closed-loop strategies with non-zero reference motion, a feasible trajectory is required. Motion planning for nonholonomic systems has been the subject of a great deal of recent research. The motion planning problem for nonholonomic vehicles can be classified by the nature of the problem, i.e. with or without obstacles or cost function to be minimized. Among the first approaches to mobile robot motion planning were the Dubin's [24], and Reeds & Shepp's [25], works. In both approaches a minimal length solution is obtained for robots in obstacle-free spaces. Also, techniques of non linear optimization can be applied (i.e. [38]). Using these approaches, the control history or trajectory is finitely parameterized. The *differential flatness* property [39], of certain nonholonomic systems, like the car-like vehicle, has been applied to the optimization strategies; for example, in Van Nieuwstadt's [40], and Milam's [41], trajectory generation algorithms. Using the flatness property of the vehicle system, the problem dimension was reduced and more computation efficiency was obtained.

2.3 Motion Coordination

The study of motion coordination of nonholonomic robot formations extends the study of motion planning of single robot. Motion coordination is a remarkable phenomenon in biological systems and an extremely useful tool for groups of vehicles. These coordination tasks must often be achieved with minimal communication between the agents and also limited information about the global state of the system. The application of these biological inspired coordination strategies should take into account the local properties of the agents, like controllability and other constraints like saturation of states and controls. Many decentralized multi-robot motion coordination research works are based in the motion of particles or the simplified models of the robot agents. In other hand, the coordination strategies which take into account the nonholonomic robot constraints often need the information of the global state of the system. We present some general strategies and analysis tools for multi-robot motion coordination, and specific applications to nonholonomic formations.

2.3.1 Artificial Potentials

Leonard and Fiorelli [42], proposed the artificial potentials which define interaction control forces between neighboring vehicles and are designed to enforce a desired inter-vehicle spacing. A virtual leader is used to influence in its neighborhood by means of additional artificial potential, then the virtual leader is used to manipulate group geometry and direct the motion of the group. This framework can be applied to homogeneous groups with no ordering of vehicles, Ozdemir and

Temeltas [2], define a turning rule of the potential forces to avoid the local minimum problems. The main drawback of these approaches, is that they are based on point mass agents and no physical or dynamical constraints are included into the model. The forces over a agent are shown in Figure 2.4

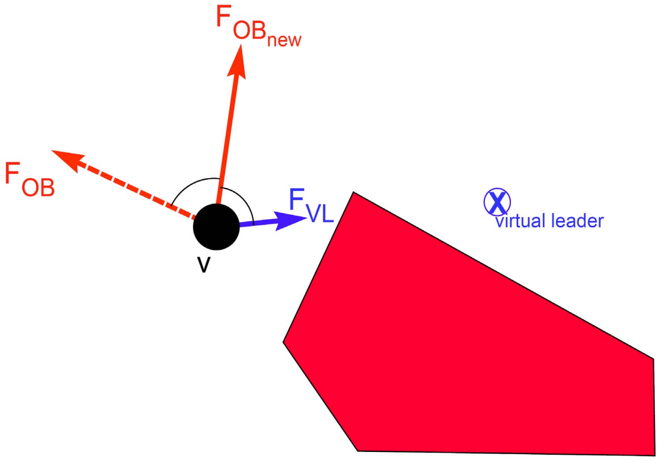


Fig. 2.4: Ozdemir and Temeltas [2], artificial potential method. Obstacle, virtual leader and modified obstacle forces definition.

2.3.2 Leader-Follower

The Leader-follower strategy was first introduced in control system by the German economist Heinrich Freiherr von Stackelberg who published *Marktform und Gleichgewicht* in 1934 which described this model (see [43]). These control methodologies, also known as Stackelberg strategies, are appropriate for classes of system problems where there are multiple criteria, multiple decision makers and decentralized information. In these strategies, the follower control actions are based on the leader's state and control. The leader-follower concept has been widely used in multi-robot control. Feddema *et al* [44], studied the observability and reachability of leader-follower based control of cooperative mobile robots. Other leader-follower applications in multi-robot systems have been presented for terrestrial vehicles,, [45], [22], [46], aerial autonomous multi-robot systems [47],[48] and unmanned underwater vehicle platoons [49],[50].

In the leader-follower approaches, each robot agent is positioned in the formation by the relative geometry with respect to its predefined neighbour. Each robot follows its predefined leader with a certain geometrical relationship. Using this leader-follower relationship, a geometrical pattern of n robots can be obtained. Typically, there is a single leader of the formation. This single leader does not follow any other robot in the set, but a predefined trajectory. The stability of leader-follower-based multi-unicycle robot systems was studied by Lechevin *et al.*, [51]. The stabilization was proposed for a formation of unicycles where the relative angle between the robots remained constant in time. Also using the leader-follower approach, Desai *et al.* [52], proposed a stabilization strategy of multiple autonomous nonholonomic robots. A framework based on graph theory for transitioning from one geometrical pattern to another was presented. Among the applications of the leader-follower strategies, the work of Dass*et al.* [22]: proposed vision-based stabilization of the formation of car-like vehicles.

2.3.3 Centralized Motion Planning

Another point of view in motion planning for multi-robot systems are the centralized planning strategies. In general, a reference trajectory is defined for the platoon then the single robot paths are computed with respect to this reference trajectory. Such of use of a reference point for all robots this strategy can be considered as a centralized approach, all robots should be able to know their distance to this reference point.

Among this strategy, Barfoot and Clark [3], propose a planning approach for mobile robot in formation, as formation they refer to certain geometrical constraints which are imposed on the relative positions and orientations of the robots throughout their travel. For platoons of nonholonomic robots keep the geometrical pattern for some manoeuvering is not possible, such of that the Barfoot and Clark work proposed to control the formation in a curvilinear coordinates rather than in the original rectilinear coordinate system. This takes advantage of the non-holonomic constraint imposed on each robot. It also ensures that for a static formation which does not turn sharper than a threshold curvature, the individual robot trajectories will not collide. They introduce an arbitrary reference point, within the formation whose coordinates serve as a single set of reference coordinates for the group. This point could be the center of the formation, one of the robots in the formation, or any other point. All robots in the formation will be described relative to this reference point (but in curved space).

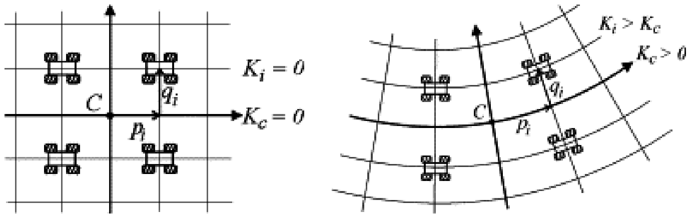


Fig. 2.5: Square and turning square formation. In curvilinear coordinates. K_C,K_i are the curvature coefficients for the reference C and robot i points. Barfoot and Clark [3]

2.3.4 Redundant Manipulator Techniques

The redundant manipulator approaches [53] have been applied by the Stilwell's [20], and Bishop's [54] works to stabilize formations of nonholonomic robots. These works were based in semi-decentralized structures. Each autonomous vehicle computes its trajectory using an exogenous feedback signal. This method was applied to formation of autonomous underwater robotic formations [55]. This strategy, to controlling a platoon of vehicles was based on the concept of controlling the platoon, not the individual vehicles. While the global behavior of the platoon is regulated, the individual vehicles behaviors are not. Indeed, the local behavior of each vehicle is not known until the closed-loop system is simulated. The goal of this approach is to regulate the platoon, thus, a suitable measure of the platoon performance is required. Any function of the platoon that can be exogenously measured will suffice. Such functions are referred to as platoon functions. In short, the Stillwel's strategy proposes a set of decentralized controllers that are implemented on each vehicle and an exogenous system that broadcasts information to the platoon, hence the approach is called semi decentralized.

They consider the the average position of the vehicles in the plane and the distribution of the vehicle positions about the average, as specific platoon functions throughout for the purposes of illustration and motivation. Overall, the control goal is to stabilize the formation about a reference trajectory of the platoon functions by controlling each vehicle velocity. As for a platoon function it can be a large number of robot configuration, the techniques of redundant manipulators are a useful stabilization strategy.

2.4 Collision-free Motion of Robots

An important task of motion planner is the navigation or the problem of finding a collision-free motion for the robot system from one configuration to another. There exists many navigation algorithms that can solve this problem. Like optimal or non-optimal solutions, online or off-line, sensor-based or world model based, etc. Optimal solutions search motions that are optimal in some way, such as distance, time, or energy. The computational complexity often depends on the memory requirements and running time of the algorithm, finally we say a planner is off-line if it constructs the plan in advance, based on know model of the environment, contrary, an online algorithm incrementally constructs the plan while the robot is executing the tasks.

2.4.1 Reactive Control

Reactive control is a term we use to describe a wide variety of schemes that have been proposed to enable robots to move without collision. Although the term is vague, what these schemes have in common is a philosophy of determining the desired motion of the robot in real time by examining some up-to-date model of the world. As the model of the world changes, the robot reacts. Typically, the model of the world is determined by the robot sensors. Also, the model may be local in that it is a function only of the current sensor information and does not contain global state that is determined over time. Reactive control goes under many names such as reactive behaviors, behavior-based control, sensor-based control, and local collision avoidance. In the following, three reactive control schemes are described: boundary following, potential fields and deformable virtual zones.

2.4.1.1 Boundary Following

The Bug1 and Bug2 algorithms [56], are among the simplest sensor-based path planning approaches. These algorithms assume the robot is a point operating in the plane with with a sensor to detect obstacles. The Bug algorithms formalizes the idea of moving toward the goal and going around obstacles. Perhaps the most straight forward path planning method is to move toward the goal, unless an obstacle is encountered, in which case, circumnavigate the obstacle until motion toward the goal is once again allowable. In Bug1 algorithm, the robot drives straight to the goal, if the robot encounters an i^{th} obstacle, let $\mathbf{q}_i^H$ be the hit point; the robot then circumnavigate the i^{th} obstacle and determines the closest point to the goal, this point is called the leaving point $(\mathbf{q}_i^L)$, from $(\mathbf{q}_i^L)$ point the robot drives direct to the goal point, and reinvokes the last described behavior. The Bug2 algorithm determines the leaving points $(\mathbf{q}_i^L)$ by searching the interception of a straight line from the start point to the goal. Then, if the robot encounters any obstacle it follow the obstacle until the leaving point. The Figure (2.6) shows sketch the Bug algorithm behavior.

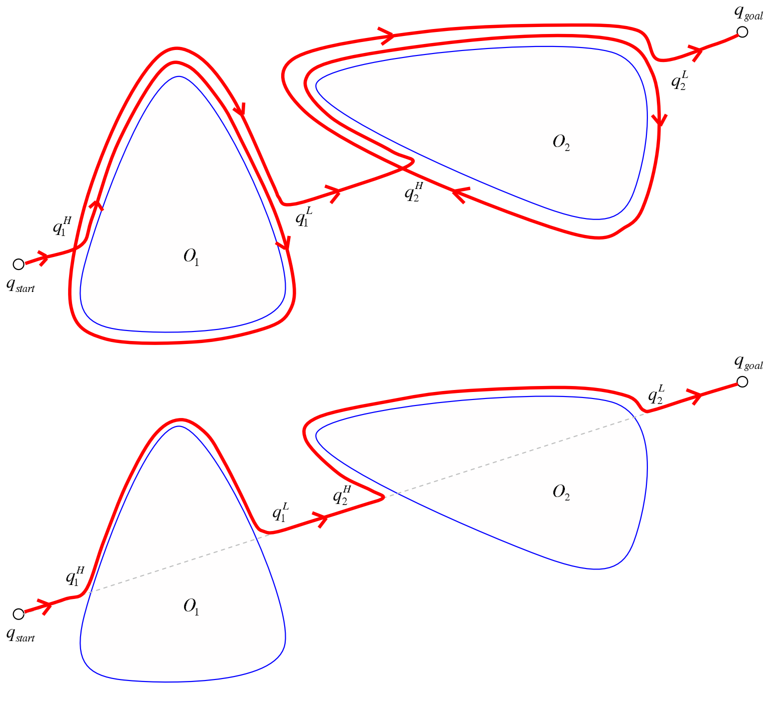


Fig. 2.6: Bug1 (up) and Bug2 (down) algorithms

Other variation of the Bug algorithm is the *Tangent Bug* algorithm [57], specifically designed for using a range sensor with a 360 degree infinite orientation resolution. These Bug algorithms are applicable only to two Degree Of Freedom (2-DOF), robots.

2.4.1.2 Potential Functions

For some motion planning problems explicitly representation of the configuration space can be difficult, such of that,an alternative is to develop search algorithms that incrementally explore free space while searching for a path. The Bug algorithms can be used for this task, but they are limited to two-dimensional configuration spaces. The *artificial potential function* method can produce a greater variety of paths than the Bug methods, and can be applied to more general class of configuration spaces, this method is based in artificial forces acting on the robot. The idea of imaginary forces acting on a robot has been suggested in 1983 by Andrews and Hogan [58] and in 1985 by Khatib [59]. In these approaches obstacles exert repulsive forces onto the robot, while the target applies an attractive force to the robot. The sum of all forces, the resultant force, determines the subsequent direction and speed of travel.

A potential field function is a differentiable real function $U : \mathbb{R}^m \rightarrow \mathbb{R}$. The value of this potential function can be interpreted as energy and hence the gradient of the potential is force. Then the gradient is a vector $\bigtriangledown U(q) = [\frac{\partial U}{\partial q_1}, \ldots \frac{\partial U}{\partial q_m}]^T$, this gradient points in the direction that locally maximally increases U. The potential function approach directs a robot as if it were a particle moving in a gradient vector field. Gradients are artificial forces acting on a positively charged particle robot which is attracted to the negatively charged goal.Obstacles have a positive charge which form a repulsive force directing the robot away from obstacles. Therefore combining the repulsive an attractive forces directs the robot from a start position to the goal while avoiding obstacles. The resulting force makes the robot follows a path descent or so called *downhill* path by following the negated gradient of the potential function. Following such a path is called *gradient descent*, i.e.:

$$\dot{\mathbf{q}} = - \bigtriangledown U(\mathbf{q})$$

Then the problem is to define additive attractive and repulsive functions, usually defined as function of the distance to the goal for the attractive forces and to the obstacles for the repulsive ones.

Koren and Borenstein [60], studied the main drawbacks of the artificial potential field methods. Among this drawbacks they cited:

Local minima or trap-situations. Perhaps the best-known and most often-cited problem with potential field methods is the problem of local minima or trap-situations. A trap-situation may occur when the robot runs into a dead end, for example a U-shaped obstacle.

No passage between closely spaced obstacles, for a closely space the repulsive forces on the robot can be pointed away from the space (see Fig. 2.8).

Also, Koren and Borenstein described oscillation conditions in narrow corridors and for the presence of large obstacles.

2.4.1.3 Deformable Virtual Zone

Although this formalism has been highly used in mobile robotics, the nonholonomy of most of them complicates the use of it. The induced kinematic constraints

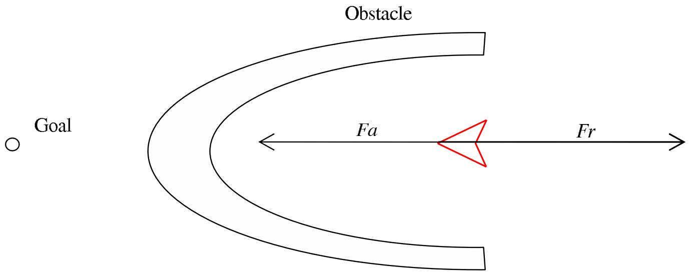


Fig. 2.7: Local minima condition in potential field methods

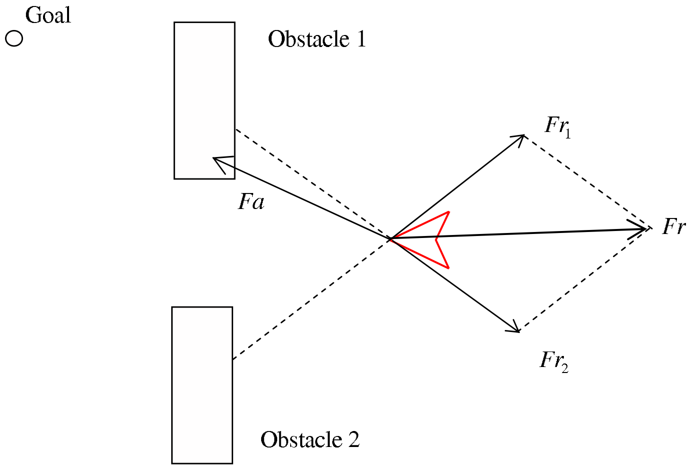


Fig. 2.8: For the potential field method the robot does not pass between closely spaced obstacles

may not allow the robot to execute instantaneously every motion, which may be lead an avoidance task to end in failure. This issue is addressed by Zapata *et al* with the DVZ approach (Deformable Virtual Zone) [4]. This consists in surrounding the robot with a virtual zone which can be deformed depending on two modes. The first one is called controlled mode. The shape of the zone is modified according to the internal state of the robot. The second mode is the uncontrolled mode of deformation. When an obstacle tries to come into the zone, it deforms its shape, as it was made of a supple membrane. The controls are computed in order to minimize this uncontrolled deformation.

2.4.2 Planning and Reactive Control

When building a robot system, we ideally would like to combine both path planning and reactive control. Path planning provides the ability to move to specified goal positions, even in the presence of complex obstacles. Reactive control provides robust performance in order to deal with uncertainties and unexpected obstacles while executing the planned path. Hence, we can say that this two concepts are excluding definitions. If we have an ideal or perfect planning solution, it will be not necessary any reaction of the system. However, to perform an ideal planning, we require to have all the information about the environment like obstacles, trajectories of moving objects etc. In real applications of robotic system, only a reduced information if often available, then an adaptive planning or combining planning and reactive control is desirable approach for some robotic application. One common approach to combining planning and reaction involves replacing paths as the specification of the planned motion of a robot. By designing a representation that reduces the level of commitment inherent in a path, a reactive controller can adapt the motion of the robot in response to information obtained during execution while still following the original plan

Some works address the adaptive planning strategies. Quinlan [61], propose a real time modification of robot collision-free paths. This method uses the elastic bands approach [62], to planning the robot path, then this trajectory is modified by the sensor information in real-tme, the strategy was validated in a PUMA 560 manipulators.

In 2004 Lamiraux *et al* [63], propose a generic approach of path optimization for nonholonomic systems. This approach is applied to the problem of reactive navigation for nonholonomic mobile robots in obstacle environments. This is a collision-free initial path being given for a robot, and obstacles detected while following this path can make it in collision. The current path is iteratively deformed in order to get away from obstacles and satisfy the nonholonomic constraints, they use the potential field method to deform the initial trajectory based on the real-time sensor information.

2.5 Localization

In general, the methods for locating mobile robots in the real world are divided into two categories: relative positioning and absolute positioning. In relative positioning, odometry and inertial navigation (gyros and accelerometers) are commonly used to calculate the robot positions from a starting reference point at a high updating rate. Odometry is one of the most popular internal sensor for position estimation because of its ease of use in real time. In contrast, the disadvantage of odometry and inertial navigation is that it has an unbounded accumulation of errors, and the mobile robot becomes lost easily. Consequently, frequent correction based on information obtained from other sensor becomes necessary.

In other hand, absolute positioning relies on detecting and recognizing different features in the robot environment in order for a mobile robot to reach a destination and implement specified tasks. These environment features are normally divided into four types [64]:

- *Active beacons* that are fixed at known position an actively transmit ultrasonic signals for the calculation of the absolute robot position from the direction of receiving incidence;
- *Artificial landmarks* that are specially designed objects or markers placed at known locations in the environment;
- *Natural landmark* or distinctive features in the environment and can be abstracted by robot sensors

- *Environment models* that are built from prior knowledge about the environment and can be used for matching new sensor observations.

2.5.1 Absolute methods

2.5.1.1 Landmark-based Navigation

In a landmark-based navigation system, the robot relies on its onboard sensors to detect and recognize landmarks in its environment to determine its position. The navigation system depends on the kind of sensors being used, the types of landmarks and the number of landmarks available. Vision by means of cameras has been applied to localize reference point in the environment. Apart of vision other sensors have bern used in position estimation, including laser, ultrasonic beacons, GPS, and sonars. As no sensor is perfect an landmarks may change none of these method is adequate to operate autonomously in the real world [64].

Global Positioning System: Among the positioning strategies for autonomous mobile robots in outdoor application is the global positioning system (GPS), [65]. The GPS was developed in 1970 by the United States Department of Defense for military applications. In 1983, President Reagan established a horizontal accuracy of 100 metres for civil users worldwide. In 1989, a new satellite group was put into service (Group II), and in 1991 the civil application signal was intentionally degraded (selective availability, SA). Selective availability was then suppressed under the Clinton administration (1996) and the accuracy of 100 metres was thus improved by a factor of 10; that is, 10 metres of horizontal accuracy for civil applications.

The GPS consists of 24 satellites in six different orbits. Four satellites are positioned in 6 different orbits. the same orbit to assure worldwide covering. Thus, every point on the earth is visible from four to ten satellites

The GPS is composed of three subsystems: spatial (Space), terrestrial (Control) and User. The Space subsystem consists of all 24 satellites, orbiting the earth every 12 hours in six orbital planes, at an altitude of 20,200 km inclined at 550 to the equator in a sun-synchronous orbit. There are often more than 24 operational satellites as new ones are launched to replace older satellites. The orbit altitude is such that the satellites repeat the same track and configuration over any point approximately every 24 hours (4 minutes earlier every day). The satellites are oriented in such a way that from any place on earth, at any time, at least four satellites are available for navigational purposes.

The Control subsystem consists of a group of four ground-based monitor stations, three upload stations and a master control station. The master control facility is located at Schriever Air Force Base in Colorado. The monitor stations track the satellites continuously and provide data to the master control station. They measure signals from the satellites, which are incorporated into orbital models for each satellite. The master control station calculates satellite ephemeris and clock correction coefficients and forwards them to an upload station; Figure 2.9 shows the localization of the terrestrial Control subsystem. The upload stations transmit the data to each satellite at least once a day. The satellites then send subsets of the orbital ephemeris to GPS receivers over radio signals.

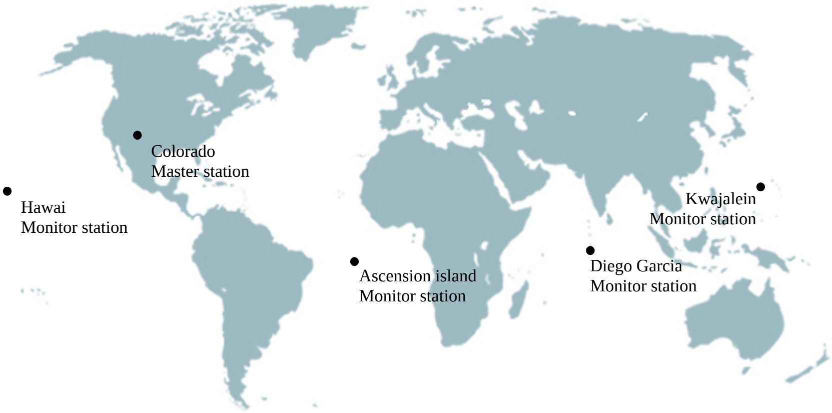


Fig. 2.9: GPS, terrestrial control subsystem stations

Finally. the GPS User Segment consists of the GPS receivers and the user community. GPS receivers convert satellite signals into position, velocity, and time estimates. Four satellites are required to compute the four dimensions: position and time.

The user position is determined by using the *pseudorange* ρ, of each satellite and its position (ephemerids) (x,y,z). The pseudorange is a measure of the time it takes the signal to leave the satellite and arrive at the user receptor. With this time and the signal velocity, the distance from the receptor to the satellite can be estimated. Each satellite has a precise atomic clock, whereas the receptor clock is conventional. Therefore, a fourth satellite has to be used for the four unknown navigation variables (user position x_u, y_u, z_u, and user clock bias b_u). For the $i^t h$ satellites, the following equation system can be written [66]:

$$\tilde{\rho}_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_u + \varepsilon_i \tag{2.4}$$

where the variable b_u is the receiver clock bias, and ε_i is the error term for the measurement. When a GPS receiver has collected range measurements from four or more satellites, it can calculate a navigation solution: (x_u, y_u, z_u) and b_u.

The sources of errors in GPS are a combination of noise and bias. The principal sources of errors in GPS are:

- Satellite clock errors uncorrected by the Control segment
- Ephemeris data errors
- Tropospheric delays

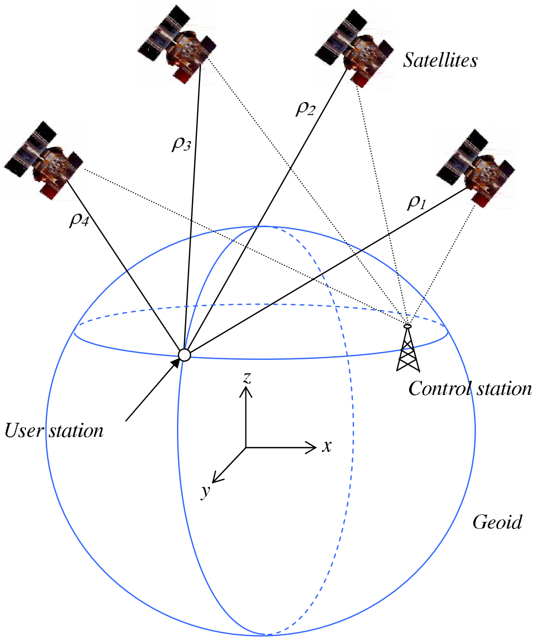


Fig. 2.10: GPS system

- Ionosphere delays
- Multipath: reflected signals from surfaces near the receiver that can either interfere with or be mistaken for the signal that follows the straight line path from the satellite

Different strategies have been proposed to reduce the errors of the GPS in mobile robot localization, i.e. [66], [67]. These works are based on the Differential GPS (DGPS), and data fusion with inertial and odometric vehicle measurements. The Extended Kalman Filter (EKF) [68], was used to fuse the data from satellite pseudoranges and inertial sensors to estimate the vehicle position.

2.5.2 Relative Methods

2.5.2.1 Odometry

Odometry is the most widely used method for determining the momentary position of a mobile robot. In most practical applications odometry provides easily accessible real-time positioning information in-between periodic absolute position measurements. Odometry is the study of position estimation during wheeled vehicle navigation. Odometry is the use of data from the rotation of wheels or tracks to estimate change in position over time, often the rotation data from the wheels are sensed by using rotary encoders. This method is often very sensitive to error. Rapid and accurate data collection, equipment calibration, and processing are required in most cases for odometry to be used effectively. For example the for a car-like robot, the problem is to estimate the position (x, y, θ) by using the angular position of the wheels and the steering angle ξ. This problem seems to be a classic direct geometric model problem for robot manipulators, where the joint space coordinates are transformed into the task space by means of the direct geometric model. However, for nonholonomic mobile robots the joint space dimension is lower than the task space. Consequently, there is not diffeomorphism between the joint and the task space; such of that, we can not use only the angular position of the robot wheels to obtain the robot position. The robot position is then estimated by integrating the angular wheels position along the vehicle trajectory [69].

2.5.2.2 Inertial Methods

Another approach to the position determination of mobile robots is based on inertial navigation with gyros and/or accelerometers. Accelerometer data must be integrated twice to yield position, thereby making these sensors exceedingly sensitive to drift. Another problem is that accelerations under typical operating conditions can be very small, on the order of 0.01 g [70]. Gyros can be more accurate (and costly) but they provide information only on the rate of rotation of a vehicle, so their data must be integrated once. This problem does not exist with electronic compasses that measure the orientation of the robot relative to the earth's magnetic field. However, electronic compasses are not recommended for indoor applications, because of the large distortions of the earth's magnetic field near power lines or steel structures.

There is two main types of gyros: piezoelectric coriolis vibrating gyros and fiber-optic gyros. In piezoelectric coriolis vibrating gyros piezoelectricity is both used to excite the in plane reference vibration and to detect the out of plane vibration induced by an input angular rate. A fibre optic gyroscope is a gyroscope that uses the interference of light to detect mechanical rotation. The sensor is a coil of as much as 5 km of optical fiber. Two light beams travel along the fiber in opposite directions. Due to the Sagnac effect [71], the beam traveling against the rotation experiences a slightly shorter path than the other beam. The resulting phase shift affects how the beams interfere with each other when they are combined. The intensity of the combined beam then depends on the rotation rate of the device.

3

Distributed Trajectory Generator

3.1 Introduction

In this chapter, we propose a distributed control strategy for formations of nonholonomic robots. Initially, we designed the general coordination strategy between the robots. This first or macroscopic level defines how the robots will interact to perform some desired group behaviour. Then, a microscopic level deals with the local mobile robot control law. The macroscopic design is a decentralized leader-follower interaction between the mobile robots. An open-loop control was then built as the local control of each robot. This local control strategy was carried out in a decentralized way using a real-time optimal trajectories generator. Finally, the stability conditions for the proposed strategy are presented.

3.2 Robots Coordination Modeling

This problem can be formulated as follows:

- Given a set $\mathscr{C}$ of $n+1$ identical nonholonomic robots, comprising a single leader and n followers.
- Given also, the initial positions and orientations of the $n+1$ robots.
- Generate a continuous trajectory for $\mathscr{C}$ subject to geometrical and kinematical constraints.

The geometrical constraints are due to the formation pattern or the relative positions between the robots, and the kinematics are mainly nonholonomic constraints.

Using the graph representation, the leader robot corresponds to node 0, whereas the followers coincide with the nodes from 1 to n of a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, which is defined by the group of vertices $\mathscr{V} = \{1, ...n\}$ and the edges $\mathscr{E} = \{(i,j) : i,j \in \mathscr{V} \cup \{0\}\}$. The relative position between the robots i and j is defined by the relative distance and angle, l_{ij} and γ_{ij} respectively for $(i,j) \in \mathscr{E}$.

For example the formations *a* and *b* in Figure (3.1), can be represented respectively by the following graphs:

$$\mathscr{G}_a = (\{1,2\},\{(0,1),(0,2)\});\;\; \mathscr{G}_b = (\{1,2\},\{(0,1),(1,2)\})$$

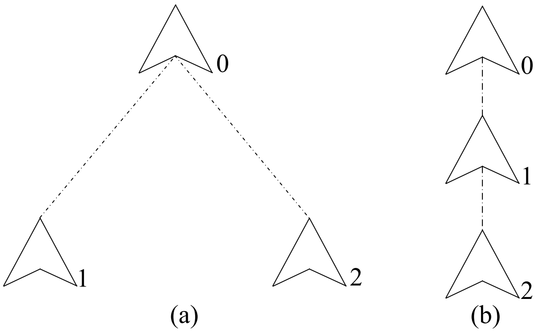


Fig. 3.1: (a) Delta robotic formation, (b) Linear robotic formation

Each vehicle is modeled as, [30]:

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{\xi}_i \end{pmatrix} = v_i \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \\ \tan\xi_i/L \\ 0 \end{pmatrix} + \omega_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{3.1}$$

where $(\; x_i \;\; y_i \;\; \theta_i \;\; \xi_i \;)$ are the planar coordinates x, y; θ is the heading angle and ξ the steering control angle (see Figure 3.2); and v_i, ω_i are the vehicle linear and steering angular velocities.

The linear and steering angular velocities $\dot{\mathbf{v}}_i = (\; v_i \;\; \omega_i \;)^T$ are subjected to the following dynamical linear model:

$$\dot{\mathbf{v}}_i = \begin{pmatrix} \dot{v}_i \\ \dot{\omega}_i \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau_v} & 0 \\ 0 & -\frac{1}{\tau_\omega R} \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} + \begin{pmatrix} \frac{1}{\tau_v} \\ \frac{1}{\tau_\omega} \end{pmatrix} \begin{pmatrix} u_{1i} \\ u_{2i} \end{pmatrix}. \tag{3.2}$$

In this system, u_{1i}, and u_{2i}, are the vehicle controls and R the radius of the vehicle's wheels. Finally, we can define the i^{th} vehicle state $\mathbf{q}_i$, as:

$$\mathbf{q}_i = (\; x_i \;\; y_i \;\; \theta_i \;\; \xi_i \;\; v_i \;\; \omega_i \;)^T \tag{3.3}$$

3.2.1 Formation Topology

The node $\mathscr{E}_{ij} = (i, j)$ defines a *leader-follower* neighbour interaction between the robot leader i and the follower j, for $i \in \mathscr{V}$.

To define the multi-robot formation, we make the following assumptions:

- The single leader of the formation tracks an exogenous generated trajectory (x^*, y^*). Hence, a non-robotic system can be used as single leader, like a human or other non autonomous system.

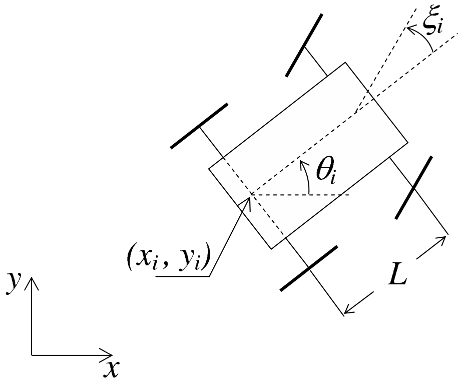


Fig. 3.2: Car-like vehicle model

- Every robot i with $i \in \mathscr{V}$, in the formation is a follower, and at least one robot is follower of the single leader of the formation, robot 0.
- The leader state $\mathbf{q}_i$, is available to robot $j \in \mathscr{V}$. We assume that the information is broadcasted by a wireless communication systems.
- The desired geometric pattern of the formation is defined by the relative distances l_{ij} and angles γ_{ij} at each node, the Figure 3.3 shows the relative positioning of the follower j.

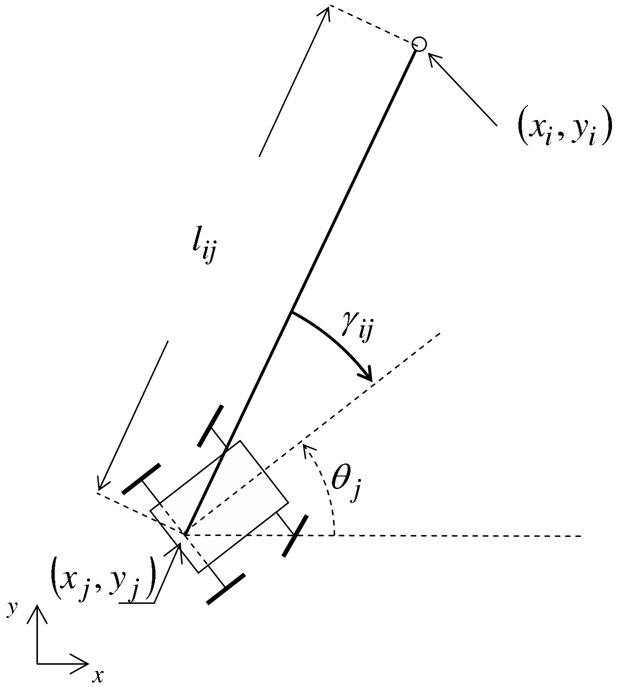


Fig. 3.3: Relative distance l_{ij}, and angle γ_{ij}.

3.3 Open-loop Control Problem

The control problem is to stabilize every formation robot j in the desired relative position l_{ij}, γ_{ij} with respect to its designated leader i where $i, j \in \mathscr{V}$. In this section we propose an open-loop control strategy to stabilize the followers in the desired l_{ij}, γ_{ij} relative positions.

Given the linear and angular velocity of the follower robots v_j and ω_j, and the leader linear velocity v_i, (see Fig. 3.4); the leader-follower distance variation $\dot{l}_{ij}$, can be expressed as:

$$\dot{l}_{ij} = -v_j \cos\left(\theta_j - \beta_{ij}\right) + v_i \cos\left(\theta_i - \beta_{ij}\right), \tag{3.4}$$

for the angle $\beta_{ij} = \theta_i - \gamma_{ij}$, and θ_i the leader orientation with respect to the inertial reference axes x, y.

We define β_{ij}, as the angle between the leader-follower direction and the inertial reference axis. Its variation, $\dot{\beta}_{ij}$, can be written as:

$$\dot{\beta}_{ij} = -v_j \frac{\sin\left(\theta_j - \beta_{ij}\right)}{l_{ij}} + v_i \frac{\sin\left(\theta_i - \beta_{ij}\right)}{l_{ij}}, \tag{3.5}$$

Finally the variations $\dot{\theta}_j$, and $\dot{\xi}_j$ are obtained from the car-like model (3.1).

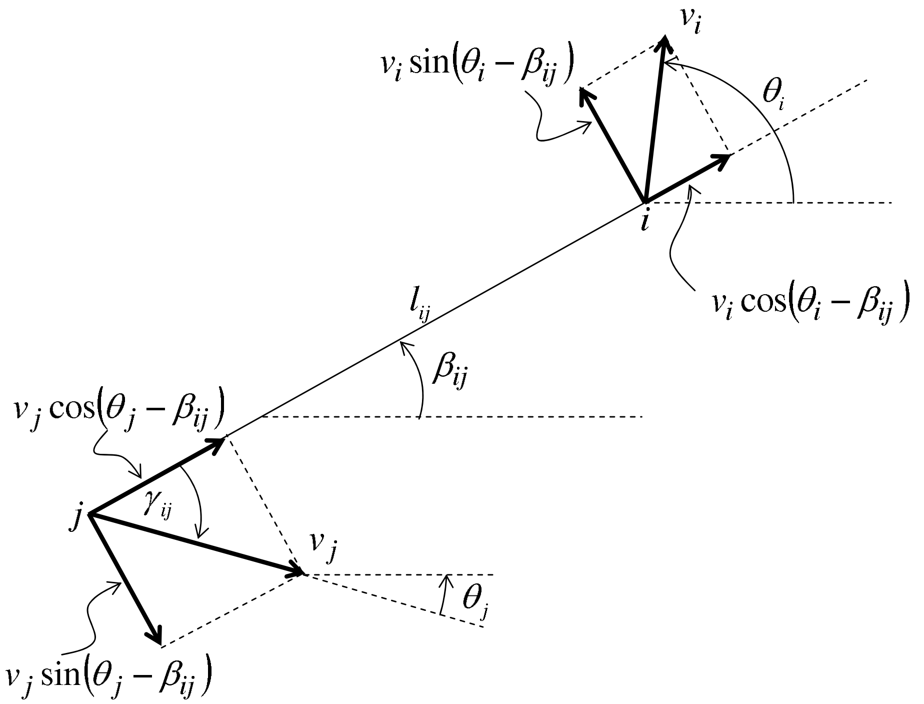

v_i
$v_i \sin(\theta_i - \beta_{ij})$
θ_i
i
$v_i \cos(\theta_i - \beta_{ij})$
l_{ij}
β_{ij}
$v_j \cos(\theta_j - \beta_{ij})$
j
γ_{ij}
v_j
θ_j
$v_j \sin(\theta_j - \beta_{ij})$

Fig. 3.4: Relative distance l_{ij}, and angle γ_{ij} variations.

Thus, for $\gamma_{ij} = \theta_j - \beta_{ij} \Rightarrow \dot{\gamma}_{ij} = \dot{\theta}_j - \dot{\beta}_{ij}$; the leader-follower interaction can be modeled as:

$$\dot{\mathbf{r}}_{ij} = \begin{pmatrix} -\cos(\gamma_{ij}) & 0 \\ \sin(\gamma_{ij})/l_{ij} + \tan(\xi_j)/L & 0 \\ \tan(\xi_j)/L & 0 \\ 0 & 1 \\ -1/\tau_v & 0 \\ 0 & -1/(\tau_\omega R) \end{pmatrix} \mathbf{v}_j + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/\tau_v & 0 \\ 0 & 1/\tau_\omega \end{pmatrix} \mathbf{u}_j + \\ ... + v_i \begin{pmatrix} \cos(\theta_i - \beta_{ij}) \\ -\sin(\theta_i - \beta_{ij})/l_{ij} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{3.6}$$

where $\mathbf{r}_{ij} = \begin{pmatrix} l_{ij} & \gamma_{ij} & \theta_j & \xi_j & v_j & \omega_j \end{pmatrix}^T$, is defined as the relative state between the robots i and j. Simplifying, the leader-follower state equation 3.6, will be expressed as:

$$\dot{\mathbf{r}}_{ij} = \mathbf{f}_{ij}\left(\mathbf{r}_{ij}, \mathbf{u}_j\right) + \tilde{\mathbf{f}}_{ij}(\mathbf{r}_{ij}), \tag{3.7}$$

If we assume a locally based velocities for each follower j, $\mathbf{v}_j = \left(v_j, \omega_j\right)$, the leader interaction function $\tilde{\mathbf{f}}_{ij}$ will be considered as an external perturbation. Thus, the control strategy is to compensate the state $\mathbf{r}_{ij}$ for any leader movements by using the local bounded controls:

$$(u_{1min}, u_{2min}) \leq (u_{1j}, u_{2j}) \leq (u_{1max}, u_{2max}). \tag{3.8}$$

The information available for the follower j (node j), is represented by the set:

$$\mathscr{I}_j = \left\{\mathbf{r}_{ij}, \theta_i, \mathbf{u}_j\right\}, \tag{3.9}$$

, and the node j local control: $\mathbf{u}_j = \begin{pmatrix} u_{1j} & u_{2j} \end{pmatrix}^T$

In the next section we propose a decentralized open loop control for the system (3.7):

$$\mathbf{u}_j = \mathbf{F}_j(\mathscr{I}_j, t), \tag{3.10}$$

to stabilize the robot j relative position into the formation, l_{ij}, γ_{ij} for $i, j \in \mathscr{V}$ and the pairs $(i, j) \in \mathscr{E}$. This approach is based in a distributed trajectory generation, with each robot computing an optimal trajectory to its desired position in the robot formation.

3.4 Optimal Trajectory Definition

There are many feasible trajectories that can join the n follower states $\mathbf{r}_{ij}$ to its desired state in the formation $\mathbf{r}_{ij}^d$. For any instant t_k an optimal trajectory $S_{kj} = \left\{\mathbf{r}_{ij}(t) \in \mathbb{R}^6 : t \in [t_k, t_{f_k}]\right\}$, and the control series $U_{kj} = \left\{\mathbf{u}_j(t) \in \mathbb{R}^2 : t \in [t_k, t_{f_k}]\right\}$, which minimizes any performance index $J_j(\mathbf{r}_{ij})$, can be computed. The time $t_{f_k} > t_k$ is the final time for the optimal control problem. This optimal problem is stated as:

$$\min_{\mathbf{r_{ij}},\mathbf{u_j}} J_j = \int_{t_k}^{t_{f_k}} L\left(\mathbf{r}_{ij},\mathbf{u}_j\right) dt, \tag{3.11}$$

where L is a scalar function, and subject to the differential constraints (eq. 3.7):

$$\dot{\mathbf{r}}_{ij} = \mathbf{f}_{ij}\left(\mathbf{r}_{ij},\mathbf{u}_{ij}\right) + \tilde{\mathbf{f}}_{ij}(\mathbf{r}_{ij}),$$

the prescribed initial conditions

$$\mathbf{r}_{ij_0} = \mathbf{r}_{ij}(t_k), \tag{3.12}$$

the prescribed final conditions, is the desired state at time t_k, which is:

$$\mathbf{r}_{ij_f} = \mathbf{r}_{ij}(t_{f_k}) = \mathbf{r}^d_{ij}(t_k), \tag{3.13}$$

the control inequality constraint

$$(u_{1min}, u_{2min}) \leq (u_{1j}, u_{2j}) \leq (u_{1max}, u_{2max}), \tag{3.14}$$

and the state inequality constraint

$$-\xi_{max} \leq \xi_j \leq \xi_{max}, \tag{3.15}$$

where xi_{max} is the maximal allowed steering angle of the vehicle's front wheels.

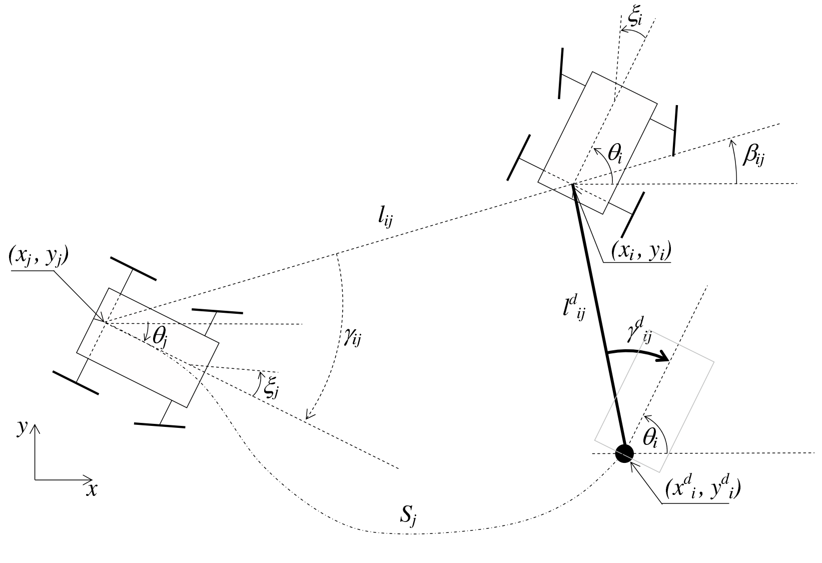


Fig. 3.5: Optimal trajectory definition

If there is an optimal solution S_{kj} at time t_k, we can feedforward this trajectory to control the robot. Then, at time t_{k+1} the initial and final constraints (eq. 3.13) are updated for the new formation configuration, then, a new optimal trajectory S_{k+1j}, is computed at time t_{k+1}. By updating the open loop trajectory each time τ where $t_{k+1} = t_k + \tau$ for $k = 1, 2, ...$, we feedback the followers relative positions, hence a closed loop trajectory planner is obtained (Fig. 3.6). This planner compensates the trajectories for variations on the relative robot positions. In the next section, the convergence condition and our study of this trajectory generator are presented.

Typically, non analytical solutions for the proposed optimal problem are available. Thus, the solution has to be obtained by a time-consuming computation numerical method. The next subsection presents an approach to applies in real time the proposed distributed trajectory planner for numerical optimization methods.

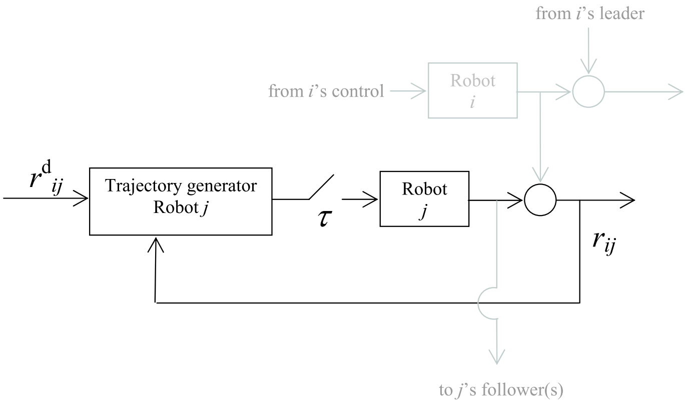


Fig. 3.6: Decentralized trajectory planner for robot follower j. with $(i,j) \in \mathscr{E}$

3.4.1 Real-time Trajectory Planning

To deal with the computation time and build the optimal reference trajectory, we use a time management strategy inspired on the Milam's [41], and Van Nieuwstadt & Murray [40] works. Under this strategy, a nominal equilibrium trajectory, denoted as S_{0j}, is used prior to any computed optimization. Then for any time t_k, we compute the optimal trajectory S_{kj}, from the initial condition at $t_k+\tau$ to the desired leader-follower final state $\mathbf{r}^d_{ij}(t_k)$. τ is the trigger time for the generation of the optimal trajectory. The trajectory generation procedure can be stated as follows:

- Let us define the time t_k as $t_k = t_{k-1}+\tau$, with $k = 1,2...$, where τ is the trigger time for the trajectory generation.
- Then, for the interval time $[t_{k-1},t_k)$, we apply the open-loop control series U_{k-1j}, and we compute the next optimal control series $U_{kj} = \left\{\mathbf{u}(t)_j \in \mathbb{R}^2 : t \in [t_k, t_{f_k})\right\}$, and its corresponding optimal trajectory $S_{kj} = \left\{\mathbf{r}_{ij}(t) \in \mathbb{R}^6 : t \in [t_k, t_{f_k}]\right\}$, with $\mathbf{r}_{ij}(t_{f_k}) = \mathbf{r}^d_{ij}(t_{k-1})$, $\mathbf{r}_{ij}(t_k) \in S_{k-1}$, and $t_{f_k} - t_k \geq \tau$; where t_{f_k}, is the planned final time for the trajectory S_{kj}.
- For the first time interval $[t_0,t_1)$ we use a nominal control series:

  $U_{0j} = \left\{\mathbf{u}(t)_j \in \mathbb{R}^2 : t \in [t_0, t_0+\tau]\right\}$, as the open-loop control and its corresponding nominal trajectory:

$S_{0j} = \left\{ \mathbf{r}_{ij}(t) \in \mathbb{R}^6 : t \in [t_0, t_0 + \tau] \right\}$, as reference trajectory.

In this algorithm we have a time τ to compute an optimal trajectory at each instant t_k. In the next section the optimal problem that was discussed above, is implemented by means of a trajectory generator algorithm.

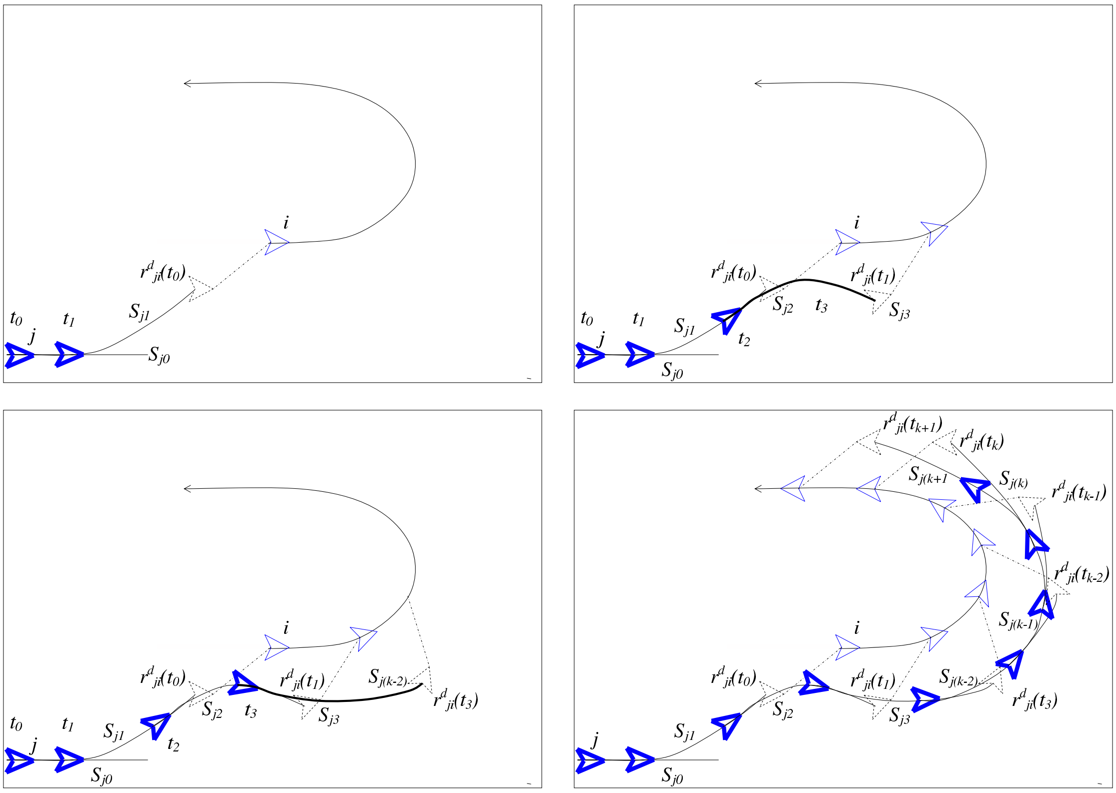


Fig. 3.7: Real time trajectory generation algorithm sequential schema

Figure (3.7) resumes the trajectory algorithm algorithm. From time t_0 to time t_1 la trajectory S_{j0} is feedwarded, also at time t_0 is started the computing of the trajectory S_{j1}, from the position $\mathbf{r}_{ij}(t_1)$ to the desired position $\mathbf{r}^d_{ij}(t_0)$. This procedure is repeated at each time t_k.

3.5 Trajectory generator Algorithm

The first step of the trajectory generation algorithm, is to map the system (3.6) outputs to a lower dimensional space, By reducing the problem dimension we can also reduce the computing time. The cost function (3.11), and the constraints (3.13,3.14), can also be mapped to this lower dimensional output space. The second step is to parameterize the outputs with a finite-dimensional approximation. Last, we transform the optimal control problem represented in the flat coordinate system into a nonlinear programming problem.

3.5.1 Robot Model and Flat Outputs

Differentially flat systems constitute a broad class of dynamical systems. They are the simplest possible extension of controllable linear systems to the nonlinear systems domain. Flat systems have a finite set of *differentially flat outputs* or outputs that do not satisfy nonlinear differential equations, so that all system variables, including the control inputs, can be exclusively written in terms of algebraic functions of such differentially independent outputs. Flat systems were first introduced by Fliess *et al*, [72], and further development and some mechanical examples were presented in the Martin's work [39].

Definition 3.5.1. *The nonlinear system* $\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u})$*, with states* $\mathbf{q} \in \mathbb{R}^n$*, is* ***differentially flat****, if there exists a change of variables* $\mathbf{z} \in \mathbb{R}^m$*, given by an equation of the form*

$$\mathbf{z} = \mathbf{h}\left(\mathbf{q}, \mathbf{u}, \dot{\mathbf{u}}, ..., \mathbf{u}^{(p)}\right), \tag{3.16}$$

such that the state and control can be determined from equation of the form:

$$\begin{aligned} \mathbf{q} &= w_s\left(\mathbf{z}, \dot{\mathbf{z}}, ..., \mathbf{z}^{(l)}\right) \\ \mathbf{u} &= w_u\left(\mathbf{z}, \dot{\mathbf{z}}, ..., \mathbf{z}^{(l)}\right) \end{aligned} \tag{3.17}$$

where $p, l \in \mathbb{N}$*. Note that* w_u, w_s *are bijective functions.*

We will refer to the change of variables $\mathbf{z}$ as the *flat outputs.*The significance of a system being flat is that all system behavior can be expressed without integration by the set of flat outputs and a finite number of its derivatives $Z = \{\mathbf{z}, \dot{\mathbf{z}}, ..., \mathbf{z}^{(l)}\}$. Then, refereing to Figure 3.8, the problem of find any trajectory that takes the nonlinear system $\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u})$, from $\mathbf{q}(0), \mathbf{u}(0)$ to $\mathbf{q}(T), \mathbf{u}(T)$, is reduced to search any smooth curve $Z(t)$ that satisfies $Z(0)$ and $Z(T)$

For the vehicle model (3.1), for any two-dimensional trajectory $z_1(t), z_2(t)$ there corresponds an unique trajectory $x(t), y(t), \theta(t), \xi(t)$, $z_1(t) = x(t); z_2(t) = y(t)$. We can see that state and input variables of system (3.1), can be obtained directly from z_1, z_2 and their successive derivatives. The states can be written as:

$$x = z_1 \tag{3.18}$$

$$y = z_2 \tag{3.19}$$

$$\theta = \arctan\frac{\dot{z}_2}{\dot{z}_1} \tag{3.20}$$

$$\xi = \arctan\left(L\frac{\dot{z}_1\ddot{z}_2 - \ddot{z}_1\dot{z}_2}{\left(\dot{z}_1{}^2 + \dot{z}_2{}^2\right)^{3/2}}\right) \tag{3.21}$$

$$v = \sqrt{\dot{z}_1{}^2 + \dot{z}_2{}^2} \tag{3.22}$$

$$\omega = \frac{\left(-3\left(\dot{z}_1\ddot{z}_2 - \ddot{z}_1\dot{z}_2\right)\left(\dot{z}_1\ddot{z}_2 + \ddot{z}_1\dot{z}_2\right) + \left(\dot{z}_1^2 + \dot{z}_2^2\right)\left(\dot{z}_1\dddot{z}_2 - \dddot{z}_1\dot{z}_2\right)\right)\sqrt{\dot{z}_1^2 + \dot{z}_2^2}L}{\left(\dot{z}_1^2 + \dot{z}_2^2\right)^3 + \left(\dot{z}_1\ddot{z}_2 - \ddot{z}_1\dot{z}_2\right)^2 L^2} \tag{3.23}$$

and the inputs:

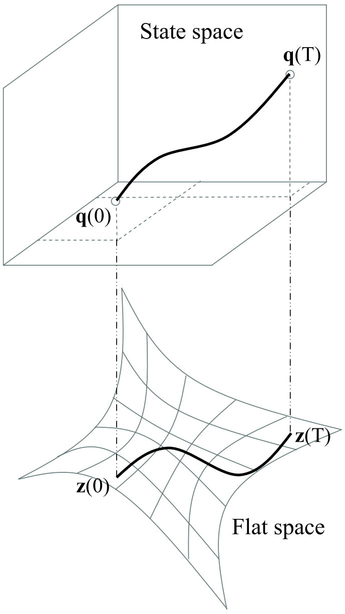


Fig. 3.8: Differentially flat system map.

$$u_1 = \frac{\tau_v\left(\dot{z}_1\ddot{z}_1 + \dot{z}_2\ddot{z}_2\right) + \dot{z}_1 + \dot{z}_2{}^2}{\sqrt{\dot{z}_1{}^2 + \dot{z}_2{}^2}} = w_1\left(\dot{z}_1, \dot{z}_2, \ddot{z}_1, \ddot{z}_2\right) \tag{3.24}$$

$$u_2 = \frac{d}{dt}\left(\omega\right)\tau_\omega + \frac{\omega}{R} = w_2\left(\dot{z}_1, \dot{z}_2, \ddot{z}_1, \ddot{z}_2, \dddot{z}_1, \dddot{z}_2, z_1{}^{(4)}, z_2{}^{(4)}\right) \tag{3.25}$$

Form equations (3.18-3.25) we note that for any almost $\mathbb{C}^4$ smooth trajectory $(z_1(t), z_2(t)), (\dot{z}_1(t)^2 + \dot{z}_2(t)^2 \neq 0)$ in the space is an admissible path for the nonholonomic system (3.1). Hence, path planning becomes easier in the flat space since we do not have to take into account any kinematic constraint along the path.

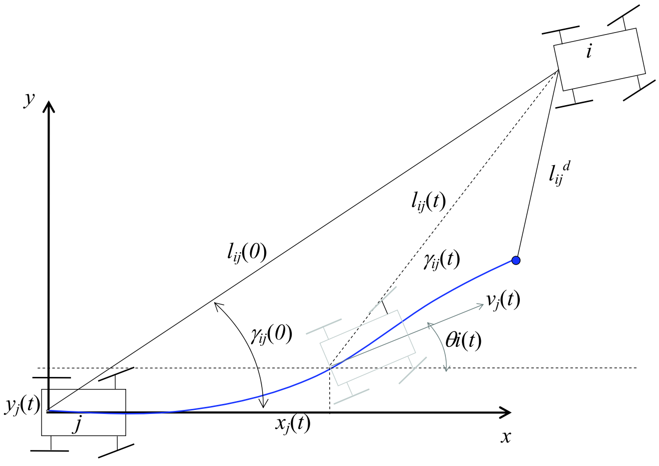


Fig. 3.9: Leader-follower relative positioning

We are interested in the relative motion between the leader and follower robot $i, j \in \mathscr{E}$ modeled by equation (3.6). Given an inertial referential coordinate system

x, y, aligned with the follower j for the initial time $t=0$, $(\theta_j(0)=x_j(0)=y_j(0)=0)$ (see Fig. 3.9); and the initial relative distance and angle $l_{ij}(0), \gamma_{ij}(0)$, the relative position between the i, j robots can be expressed as a function of the flat outputs by the following expressions:

$$l_{ij} = \sqrt{\left(l_{ij0}\cos\gamma_{ij0} - z_1\right)^2 + \left(l_{ij0}\sin\gamma_{ij0} - z_2\right)^2} \tag{3.26}$$

$$\gamma_{ij} = \arctan\frac{\left(l_{ij0}\sin\gamma_{ij0} - z_2\right)}{\left(l_{ij0}\cos\gamma_{ij0} - z_1\right)} - \arctan\frac{\dot{z}_2}{\dot{z}_1}, \tag{3.27}$$

Finally, by representing the system by its flat output $Z = \left(\dot{z}_1, \dot{z}_2, \ddot{z}_1, \ddot{z}_2, \dddot{z}_1, \dddot{z}_2, {z_1}^{(4)}, {z_2}^{(4)}\right)$, we can rewrite the optimal problem (3.11) as:

$$\min_{\mathbf{z_j}} J_j(Z_j, t). \tag{3.28}$$

With the constraints (3.13-3.15) also expressed as a function of the flat output $\mathbf{z}$:

3.5.2 Trajectories with B-splines Parameterization

In the previous section, techniques were presented to reduce or eliminate the dynamic constraints by selecting a special set of variables (outputs) that could completely characterize the states and inputs of the system under consideration. In this section, we will discuss how to select the outputs from a finite dimensional space so that the problem under consideration can be efficiently solved.

There are many curves that can be used to approximate the outputs (Fourier series, polynomials, rational segments, etc.). Aside from accurately representing a basis of the solution of the trajectory generation problem under consideration with a reasonable number of decision variables, the main requirements of the curve are the ability to set a level of continuity $\mathscr{C}^k$, without adding additional constraints. Specifying the level of continuity is necessary, since the states and inputs are a function of the outputs and their derivatives. The B-spline polynome mets this continuity constraint with a numerically stable computer implementation. An overview of B-splines, from which much of the following is derived, can be found [73].

The system flat output B-spline parameterization, can be defined by:

$$z_1(t) = \sum_{k=0}^{h} B_{k,r}(t) C_k^1, \qquad z_2(t) = \sum_{i=0}^{h} B_{k,r}(t) C_k^2, \tag{3.29}$$

where:

- C_k^1, C_k^2, are the free parameters or the degrees of freedom, for the flat outputs z_1, z_2 respectively

- $h+1$, the number of free parameters
- r, is the degree of the polynomial pieces
- $B_{k,r}(t)$, are the basis functions

The B-spline basis functions are defined by :

$$B_{k,0}(t) = \begin{cases} 0 \quad if & t \in [u_k, u_{k+1}) \\ 1 \quad otherwise & \end{cases}$$

$$B_{k,r}(t) = \frac{t-u_k}{u_{k+r}-u_k} B_{k,r-1}(t) + \frac{u_{k+r+1}-t}{u_{k+r+1}-u_{k+1}} B_{k+1,r-1}(t) \tag{3.30}$$

where, u_k, are the knots and $h+r$ the number of knots.

In the Figures 3.10 and 3.11, we see an example of a B-spline path parameterizations. Figure 3.10 shows the basis b-spline functions, and in Figure 3.11 a parameterized 2D-path is represented for a set of free parameters $\mathbf{C} = \{C_k^1, C_k^2\}$.

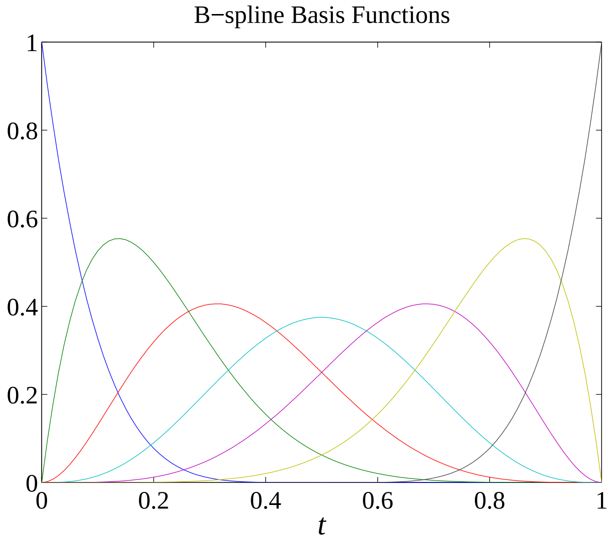


Fig. 3.10: 6 order seven B-splines basis functions $B_{k,r}(t)$

Now, the parameterized flat output n^{th} derivatives are described by:

$$z_1^{(n)}(t) = \sum_{k=0}^{h} B_{k,r}^{(n)}(t) C_k^1, \qquad z_2^{(n)}(t) = \sum_{i=0}^{h} B_{k,r}^{(n)}(t) C_k^2, \tag{3.31}$$

Hence, the flat output set $\mathbf{z}$, and consequently the system state $\mathbf{r}$ and input $\mathbf{u}$ can be described by the set of free parameters $\mathbf{C} = \{C_k^1, C_k^2\}$. Thus, the optimal control (3.28) problem is transformed into a parameter optimization problem:

$$\min_{\mathbf{C},t} F(\mathbf{C},t), \tag{3.32}$$

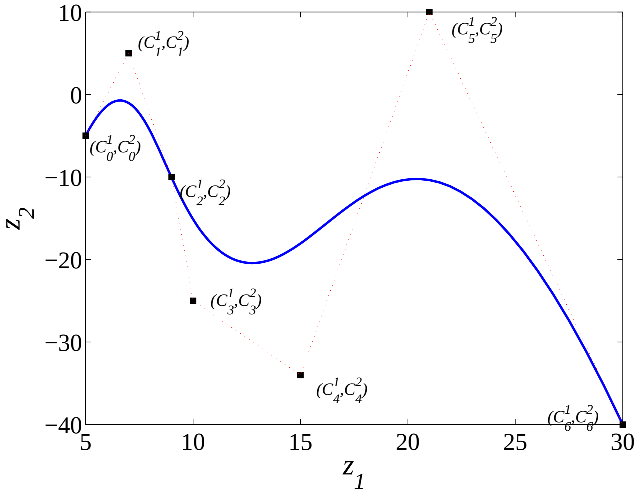


Fig. 3.11: (z_1, z_2) B-spline parameterized path

with the equality and inequality constraints:

$$L \leq \left\{ \begin{array}{c} \mathbf{C} \\ t \\ I(\mathbf{C}, t) \\ S(\mathbf{C}, t) \end{array} \right\} \leq U \qquad (3.33)$$

where F is the performance index as a function of the parameters set points, I and S, which are the trajectory constraints for the input 3.14 and states 3.15 respectively, written as function of the free parameters $\mathbf{C}$, and L and U the upper and lower bounds.

The optimal problem was defined in the flat space, then the flat space trajectories were parameterized by the free parameters of B-splines. In this way, the optimal control problem was transcribed into a nonlinear programming problem. In the next subsection, the performance index criterion is defined and an algorithm for the nonlinear programming problem is proposed.

3.5.3 Transcription into a Nonlinear Programming Problem

A reliable method to convert an optimal control problem to a nonlinear programming problem is collocation. The basis of the direct collocation approach is a finite dimensional approximation of control and state variables, i.e. a discretization. For more in direct collocation methods see [74],[75]

First we break the time domain into smaller intervals:

$$t_0 < t_1 < t_2 < ... t_N = t_f,$$

The nonlinear programming decision variables Y then become the values of the state and the control at the grid points, namely:

$$Y = \{Z(t_1), Z(t_2), ..., Z(t_{N-1}), Z(t_N)\}\,. \tag{3.34}$$

The collocation points are the points in the time interval that the constraints are enforced. The integration points or mesh points must also be specified. Without loss of generality, we assume that the integration points are identical to the collocation points and the same for each constraint.

Hence, the const function J_j, can be approximated as follows:

$$J_j \approx \sum_{l=0,l=l+P}^{N} \sum_{k=0}^{P} \mu_k L\left(w_s(Z_{k+l}), w_u(Z_{k+l}), t_{k+l}\right), \tag{3.35}$$

the term μ_k is the quadrature term for the integral approximation. The N value is the number of collocation points, and P depends of the quadrature rule, for example for the Simpson rule, we have $P = 1$ (see Fig. 3.12).

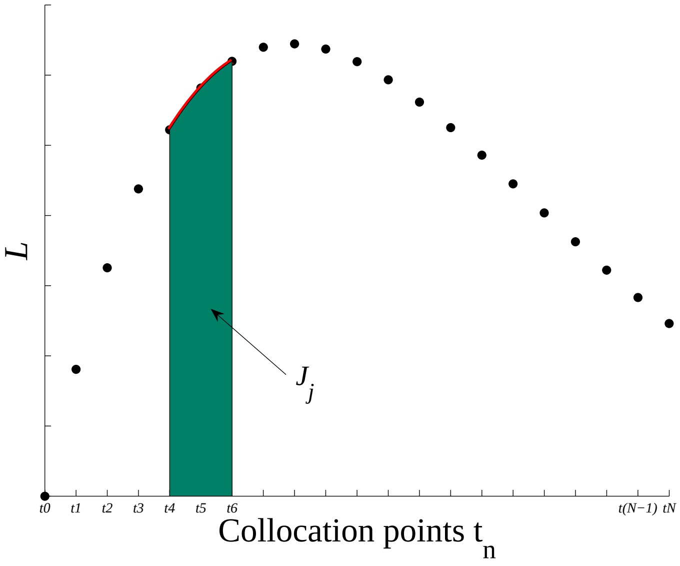


Fig. 3.12: Hypothetical performance index approximation by the Simpson rule ($P = 2$) , This figure shows the J_j approximation for the collocation points $t_l, t_{l+P} = t_4, t_6$

Figure(3.13), illustrates the transcription process, from the optimal control problem to nonlinear programming.

3.6 Time-Optimal Trajectories

We were interested in a minimal time path between the initial condition and the final configuration for the robot follower j. The initial condition is the current robot j position $\mathbf{r}_{ij}$ and the final configuration $\mathbf{r}^d_{ij}$, is determined by the leader-follower desired relative position.

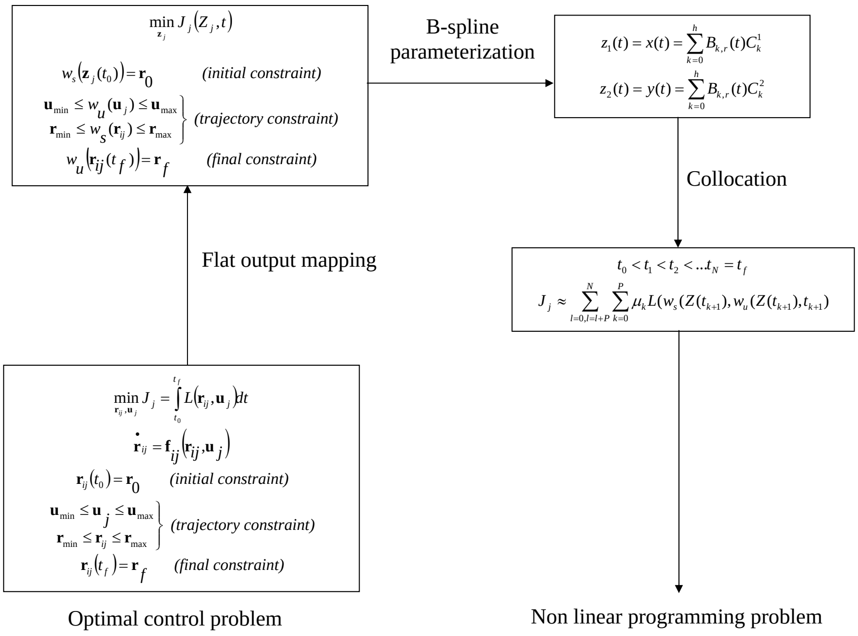

$\min_{\mathbf{z}_j} J_j(Z_j, t)$
$w_s(\mathbf{z}_j(t_0)) = \mathbf{r}_0$ (initial constraint)
$\mathbf{u}_{\min} \le w_u(\mathbf{u}_j) \le \mathbf{u}_{\max}$
$\mathbf{r}_{\min} \le w_s(\mathbf{r}_{ij}) \le \mathbf{r}_{\max}$ (trajectory constraint)
$w_u(\mathbf{r}_{ij}(t_f)) = \mathbf{r}_f$ (final constraint)
B-spline parameterization
$z_1(t) = x(t) = \sum_{k=0}^{h} B_{k,r}(t) C_k^1$
$z_2(t) = y(t) = \sum_{k=0}^{h} B_{k,r}(t) C_k^2$
Collocation
$t_0 < t_1 < t_2 < \ldots t_N = t_f$
$J_j \approx \sum_{l=0,l=l+P}^{N} \sum_{k=0}^{P} \mu_k L(w_s(Z(t_{k+1}), w_u(Z(t_{k+1}), t_{k+1})$
Flat output mapping
$\min_{\mathbf{r}_{ij}, \mathbf{u}_j} J_j = \int_{t_0}^{t_f} L(\mathbf{r}_{ij}, \mathbf{u}_j) dt$
$\dot{\mathbf{r}}_{ij} = \mathbf{f}_{ij}(\mathbf{r}_{ij}, \mathbf{u}_j)$
$\mathbf{r}_{ij}(t_0) = \mathbf{r}_0$ (initial constraint)
$\mathbf{u}_{\min} \le \mathbf{u}_j \le \mathbf{u}_{\max}$
$\mathbf{r}_{\min} \le \mathbf{r}_{ij} \le \mathbf{r}_{\max}$ (trajectory constraint)
$\mathbf{r}_{ij}(t_f) = \mathbf{r}_f$ (final constraint)
Optimal control problem
Non linear programming problem

Fig. 3.13: Transcription of optimal control problem to nonlinear programming problem

The objective function for this time minimization can be defined as:

$$J = \int_{t_0}^{t_f} dt = t_f - t_0. \tag{3.36}$$

if we define $t_0 = 0$,then:

$$J = t_f. \tag{3.37}$$

Since the final time is unknown, it must be a parameter of optimization. Thus, we define the normalized time $\tau = t/t_f$, where t_f is the unknown constant. The time derivatives are scaled as: $\frac{d}{dt} = \frac{1}{t_f}\frac{d}{d\tau}$. The normalized problem is then described as:

$$\min t_f \tag{3.38}$$

subject to the initial conditions:

$$\begin{array}{rcl} z_1(0) & = & x_0 \\ z_2(0) & = & y_0 \\ \frac{\sqrt{\dot{z}_1^2(0)+\dot{z}_2^2(0)}}{t_f} & = & V_0 \\ \arctan\frac{\dot{z}_2^2(0)}{\dot{z}_1^2(0)} & = & \theta_0 \\ \arctan\left(L\frac{\dot{z}_1(0)\ddot{z}_2(0)-\ddot{z}_1(0)\dot{z}_2(0)}{\left(\dot{z}_1{}^2(0)+\dot{z}_2{}^2(0)\right)^{3/2}}\right) & = & \xi_0 \end{array}, \tag{3.39}$$

the final conditions

$$\begin{array}{rcl} z_1(t_f) & = & x_f \\ z_2(t_f) & = & y_f \\ \frac{\sqrt{\dot{z}_1^2(t_f)+\dot{z}_2^2(t_f)}}{t_f} & = & V_f \end{array}, \tag{3.40}$$

and the trajectory conditions:

$$\begin{array}{rcccl} V_{min} & \leq & \frac{\sqrt{\dot{z}_1^2(\tau)+\dot{z}_2^2(\tau)}}{t_f} & \leq & V_{max} \\ \xi_{min} & \leq & \arctan\left(L\frac{\dot{z_1}(\tau)\ddot{z_2}(\tau)-\ddot{z_1}(\tau)\dot{z_2}(\tau)}{\left(\dot{z_1}^2(\tau)+\dot{z_2}^2(\tau)\right)^{3/2}}\right) & \leq & \xi_{max} \end{array}, \tag{3.41}$$

The flat outputs $z_1(\tau), z_2(\tau)$ are parameterized using the above defined b-splines curves (eq. 3.29), for the the normalized time $\tau \in [0,1]$. Seven B-splines of 6^{th} order were used to parameterized the system flat outputs. Therefore, each B-spline is characterized by a set of seven free parameters $\mathbf{C} = \{C_k^1, C_k^2\}, k = 7$.

NPSOL [76], SNOPT [77] and CFSQP [78] are among the most often used nonlinear programming solvers. SPNOT uses a sequential quadratic programming (SQP) algorithm. Search directions are obtained from QP subproblems that minimize a quadratic model of the Lagrangian function subject to linearized constraints. An augmented Lagrangian merit function is reduced along each search direction to ensure convergence from any starting point. NPSOL is a set of Fortran subroutines that also uses sequential quadratic programming (SQP) and merit functions. Finally, CFSQP is a set of C functions for the minimization of the maximum of a set of smooth objective functions (possibly a single one, or even none at all) subject to general smooth constraints. If the initial guess provided by the user is infeasible for some inequality constraint or some linear equality constraint, CFSQP first generates a feasible point for these constraints; subsequently all the successive iterates generated by CFSQP satisfy these constraints. For more deepest discussion on nonlinear programming see [79].

We used the CFSQP solver to obtain the solution of the nonlinear programming problem 3.38. This solver permitted us to obtain a solution that satisfies the problem constraints at each iteration, i.e. a feasible reference trajectory will be available at each iteration, and a suboptimal feasible reference trajectory will be available even if no optimal solution is obtained within the sample time of the real time trajectory planning described in section 3.4.1.

In the Figures (3.14 to 3.18) the simulation for a formation of three robots are shown. The two followers (1 and 2) follow the arbitrary leader's trajectory. Each figure contrast the solution for two different sampling times ($T_s = 300ms$ and $T_s = 500ms$). That is, each optimal trajectory is computed within the time T_s. If no optimal solution is obtained the last suboptimal path is used as referential trajectory.

From the Figures 3.15, 3.16, we can see how the planned velocity and steering angle saturation limits are respected. The value for velocity saturation is $6m/s$ and the maximal and minimal steering angles are $\frac{\pi}{6}rad$.

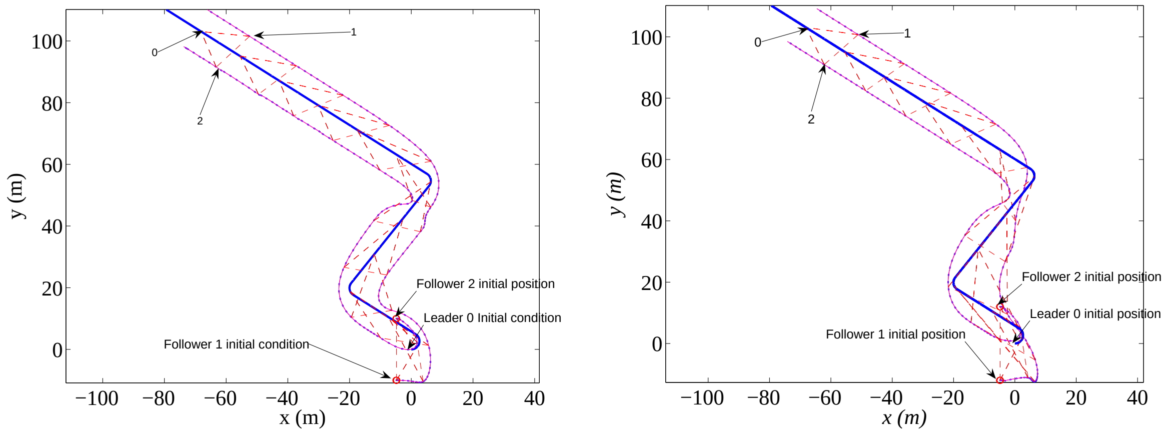


Fig. 3.14: Formation trajectory: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Initial leader position$= (0,0)$. Bounded computing time: left figure $T_s = 300ms$; right figure $T_s = 500ms$

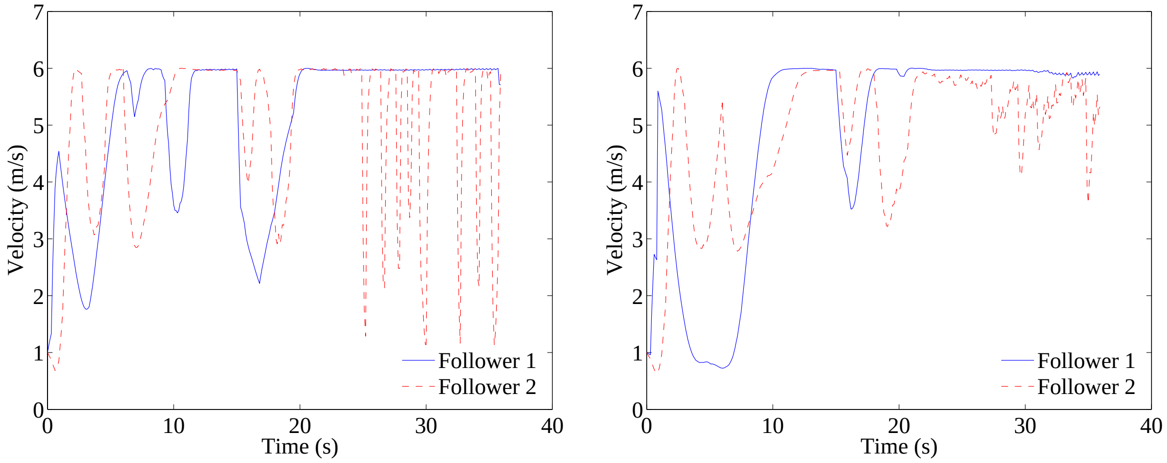


Fig. 3.15: Linear velocities v_i: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Initial leader position$= (0,0)$. Bounded computing time: left figure $T_s = 300ms$; right figure $T_s = 500ms$

Finally, for this simulation the Figures 3.17 and 3.18 show the values of the relative distances and angles $l_{ij}, \gamma ij$. In these two graphics we see how the relative distances l_{ij} and angles γ_{ij} are stabilized about the desired values, $(10m)$ and $(45°, -45°)$ respectively.

The obtained trajectory for $T_s = 500ms$ differs from the trajectory for $T_s = 300ms$ (Fig. 3.14). For the $T_s = 300ms$ simulation, the stabilization time about the desired distances and angles (see Fig. 3.17 and 3.18) is greater than in the case of $T_s = 500ms$. By limiting the computing time (from 500ms to 300ms), only a suboptimal trajectory can be computed.

A formation of thirteen robots is shown in figure 3.19, in this simulation the

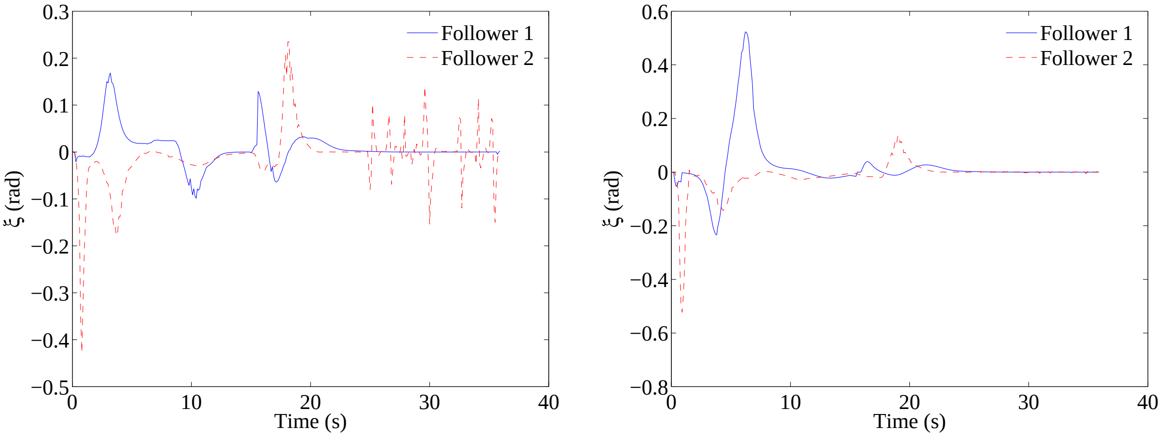


Fig. 3.16: Steering angle ξ_i: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Initial leader position= $(0,0)$. Bounded computing time: left figure $T_s = 300ms$; right figure $T_s = 500ms$

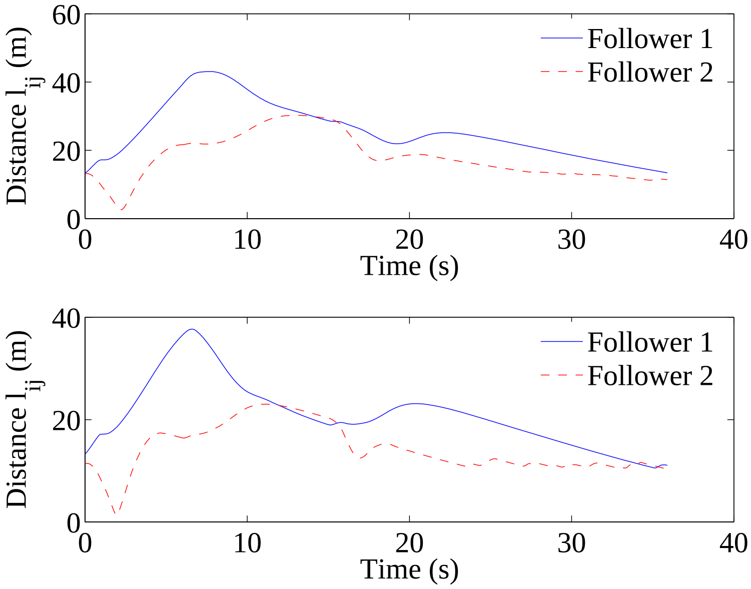


Fig. 3.17: Relative distance l_{ij}: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Initial leader position= $(0,0)$. Bounded computing time: top figure $T_s = 300ms$; bottom figure $T_s = 500ms$

robot geometrical pattern is changed in time $t = 11s$, we can see how the formation is stabilized, first in the delta G_1 formation, then in the composite formation G_2.

From the simulation of the proposed control strategy we can see that the robots are stabilized in the desired geometrical pattern, with a transit behaviour. This transit comportment is evident for the rotation of the leader. For any manoeuvring the velocity of the desired position can be greater than the maximal follower's velocity, in which case a transit behaviour is observed. That is, the pattern of the robot set is deformed while the leader robot turn, then the geometrical form is

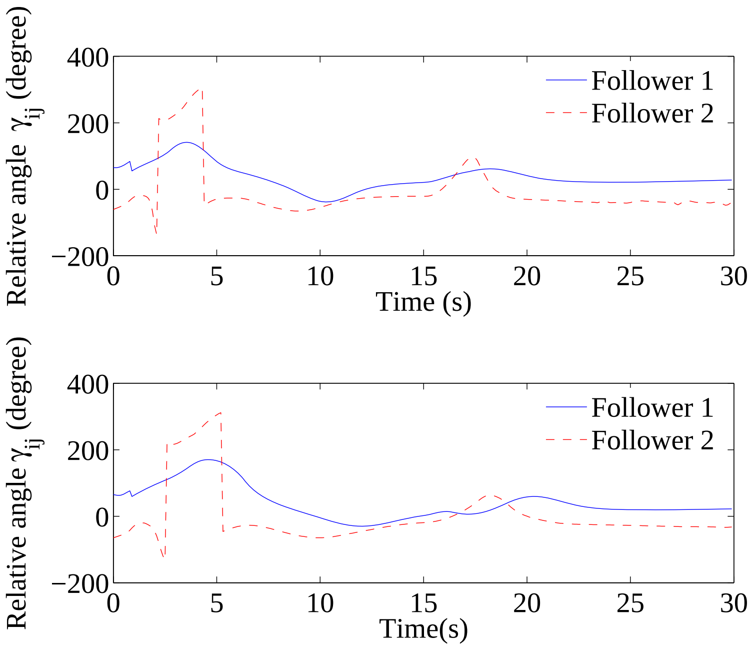


Fig. 3.18: Relative angle γ_{ij}: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Initial leader position= $(0,0)$. Bounded computing time: top figure $T_s = 300ms$; bottom figure $T_s = 500ms$

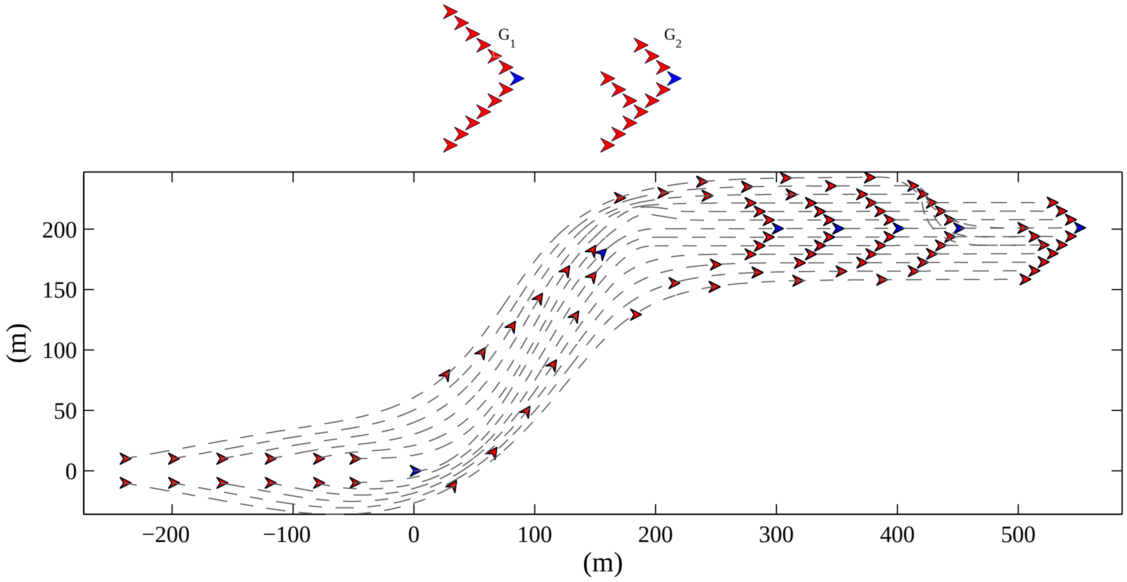


Fig. 3.19: 13 robot formation. Top: Desired formation configurations, G_1 if $t \leq 11s$, G_1 if $t > 11s$. Bottom: trajectories

.

stabilized.

In the next section this transit behaviour is studied and the condition for the convergence of the proposed trajectory generator algorithm are developed.

3.7 Convergence Conditions

To develop the convergence condition we used a unicycle model without lost of generality:

$$\dot{\mathbf{x}} = v(\ \cos\theta \quad \sin\theta\ )^T, \tag{3.42}$$

with the coordinates $\mathbf{x} = (\ x \quad y\ )^T$

We defined the trajectory generation convergence as the limiting behaviour of the trajectory generation algorithm (section 3.4.1), specifically stating that the trajectory algorithm generation converges if the planned trajectory S_k matches the leader-follower desired state in a finite time. Therefore, we can define the convergence as:

Definition: For a follower initial condition $\mathbf{x}_{foll}(t_0)$, the trajectory algorithm converges at time $t_k = t_n$ if: $\mathbf{x}_{foll}(t_n) = \mathbf{x}^d_{foll}(t_n)$. Where n is a finite number.

We propose to show the conditions of convergence for the proposed real-time trajectory generation algorithm, under the following conditions: non-obstacle environment and that the follower robot velocities are constant and equal to its maximal value. Then we can state the following proposition:

Proposition: If the trajectories S_k are feasible and time optimal, and if the trajectory of the desired state $\mathbf{x}^d_{foll}(t)$ is also in a feasible trajectory $(|v^d_{foll}(t)| < v_{max}, |\omega^d_{foll}(t)| < \omega_{max})$, then the trajectory generation algorithm converges.

Proof: Let $p_k = \{\mathbf{x}^p_{foll}(t) : t \in [t_k, t_{fk}]\}$ be a planned feasible trajectory for the robot follower, with $\mathbf{x}^p_{foll}(t_{fk}) = \mathbf{x}^d_{foll}(t_k)$, $\hat{p}_k = \{\mathbf{x}^p_{foll}(t) : t \in [t_k, t_{k+1}]\} \subseteq p_k$ be the planned trajectory subset from t_k to t_{k+1}, and finally $p^*_k = \{\mathbf{x}^d_{foll}(t) : t \in [t_k, t_{k+1}]\}$ be a feasible trajectory that joins the desired states $\mathbf{x}^d_{foll}$, at times t_k and t_{k+1}, respectively,(see Fig. 3.20).

At the time t_k we can obtain the distance d_k, which is the trajectory p_k distance between he current follower state $\mathbf{x}_{foll}(t_k)$ and the desired leader-follower state $\mathbf{x}^d_{foll}(t_k)$. This distance d_k can be expressed as:

$$d_k = d_{k-1} + \Delta d_k, \tag{3.43}$$

with

$$d_k \geq 0 \tag{3.44}$$

The variation of the distance Δd_k is given by:

$$\begin{aligned} \Delta d_k &= \int_{p^*_k} ds - \int_{\hat{p}_k} ds \\ &= \int_{t_k}^{t_{k+1}} |v_{foll}(t)| dt - \int_{t_k}^{t_{k+1}} |v^d_{foll}(t)| dt \\ &= \int_{t_k}^{t_{k+1}} \Big(|v_{foll}(t)| - |v^d_{foll}(t)| \Big)\, dt, \end{aligned} \tag{3.45}$$

where $|v_{foll}(t)|$ and $|v^d_{foll}(t)|$ are respectively the linear velocities of the robot follower and of the desired leader-follower position. If p_k is a solution with the control

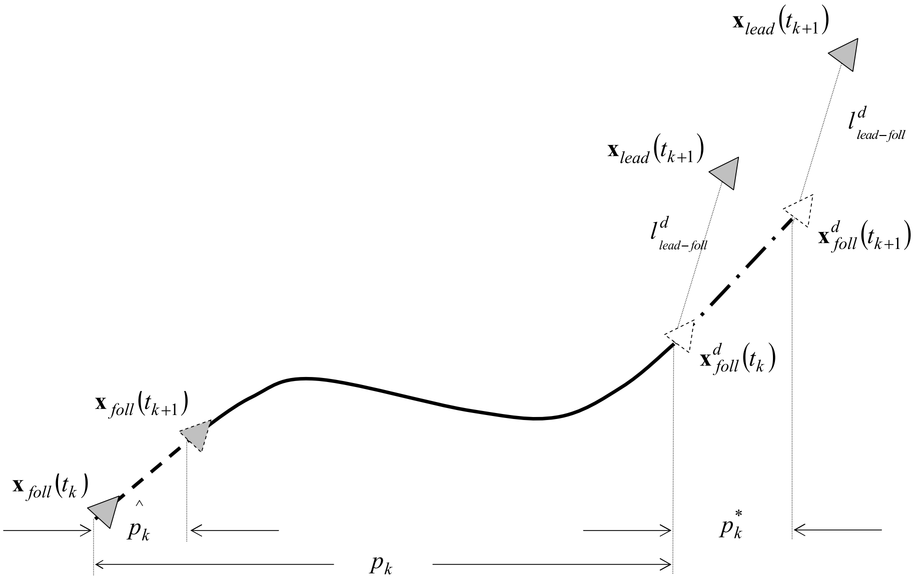

$\mathbf{x}_{lead}(t_{k+1})$
$l^{d}_{lead-foll}$
$\mathbf{x}_{lead}(t_{k+1})$
$l^{d}_{lead-foll}$
$\mathbf{x}^{d}_{foll}(t_{k+1})$
$\mathbf{x}^{d}_{foll}(t_{k})$
$\mathbf{x}_{foll}(t_{k+1})$
$\mathbf{x}_{foll}(t_{k})$
$\hat{p}_k$
p^{*}_k
p_k

Fig. 3.20: Trajectory generation convergence.

action $v_{foll}(t)$ at a bounded value v_{max}, or $\{|v_{foll}(t)| = v_{max} : \forall t \in [t_k, t_{fk}]\}$. Then as $|v^{d}_{foll}(t)| < v_{max}$:

$$\Delta d_k = \left(|v_{foll}(t)| - |v^{d}_{foll}(t)|\right)\tau < 0, \tag{3.46}$$

If $d_k \geq 0$ (eq. 3.44), and $\Delta d_k < 0$ (eq. 3.46) then:

$$\Delta d_k \, d_k \leq 0 \tag{3.47}$$

That is, $d_k \to 0$ as $k \to \infty$, or the follower trajectory will converge to the desired position in the formation when there is no obstacle to avoid.

3.7.1 Leader-Follower Convergence Conditions

In order to ensure the trajectory generation convergence, we show that the value of the linear velocity of the desired follower position $|v^{d}_{foll}(t)|$ has to be less than the maximal robot velocity value V_{max}. The aim of this section is to obtain the imposed follower desired velocities into the formation. We suppose the robot formation as a rigid body, then, by the geometrical desired pattern and the formation leader trajectory we can obtain the desired linear velocities of each follower. Finally we can obtain the speed velocity restrictions that will ensures the convergence of all follower robots to is desired positions, under any maneouvring of the leader of the formation.

To obtain the desired follower positions and their variations in time, we consider the robot formation as a rigid body, or in simpler terms, that the robot set pattern is not deformed for any formation leader movement. Then, for any formation leader displacement, we can obtain where the followers desired positions are, by means of the rigid body mechanics relationships.

Knowing, the localization of the instantaneous centre of rotation $\boldsymbol{ICR}$ for the formation leader vehicle, and the relative desired leader-follower distance l_{l-f}, and angle γ_{l-f}, we can obtain the desired follower velocity v^d_{foll}, as a function of the distance from the desired follower position to the $\boldsymbol{ICR}$ R^d, the leader velocity v_{lead}, and the angular velocity for the leader robot ω, which will be the same for the follower robots if we consider the formation as a rigid body (see Fig. 3.21.

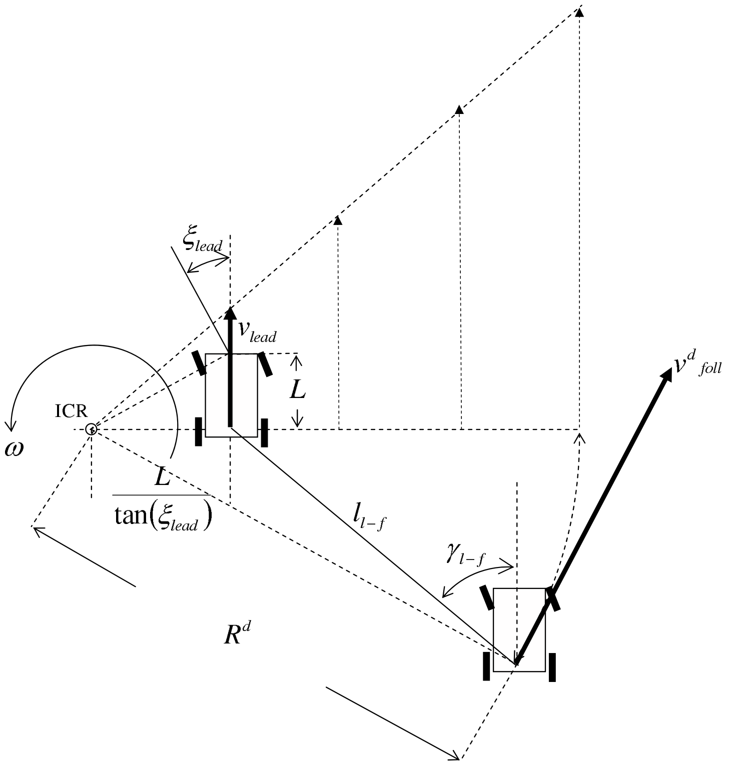


Fig. 3.21: Formation desired velocities.

Then, follower velocity can be defined as:

$$|v^d_{foll}| = R^d \omega, \tag{3.48}$$

where the angular velocity can be determined from the leader vehicle model as:

$$\omega = \frac{v_{lead} \tan \xi_{lead}}{L}, \tag{3.49}$$

and the distance from the follower to the $\boldsymbol{ICR}$ is:

$$R^d = \sqrt{\frac{L^2}{\tan^2 \xi_{lead}} + \frac{l_{l-f} L \cos \gamma_{l-f}}{\tan \xi_{lead}} + l^2_{l-f}}, \tag{3.50}$$

finally substituting equations 3.49 and 3.50 into the relation 3.48, and simplifying, we obtain the desired follower velocity, wich can be expressed as:

$$|v^d_{foll}| = |v_{lead}| \sqrt{1 + \frac{\tan^2 \xi_{lead}}{L^2} l^2_{l-f} + \frac{2 \cos \gamma_{l-f} l_{l-f} \tan \xi_{lead}}{L}} \tag{3.51}$$

Then, from the convergence conditions, the decentralized trajectory generation of each robot follower will converge if:

$$|v_{lead}| < \left(\frac{v_{max}}{\sqrt{1 + \frac{\tan^2 \xi_{lead}}{L^2} l^2_{(l-f)} + \frac{2\cos\gamma_{(l-f)} l_{(l-f)} \tan\xi_{lead}}{L}}} \right), \tag{3.52}$$

for all the followers where l_{l-f} and γ_{l-f} are, respectively, the relative distance and angle between the leader robot and the follower (see Fig. 3.22).

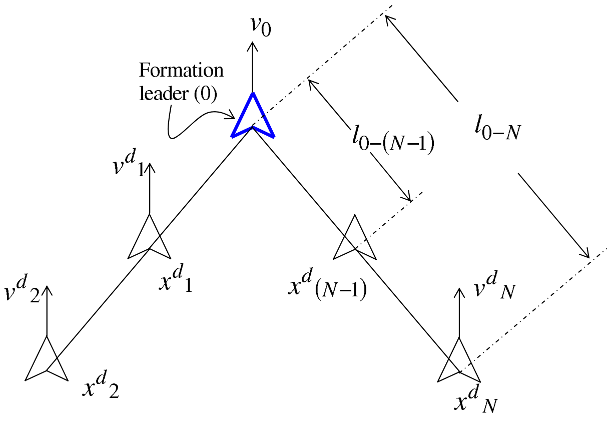


Fig. 3.22: Leader-Follower formation configuration.

That is, if we know the relative desired position l, and γ with respect to the absolute leader,we can determine by using (3.52) the maximal velocity v_{lead} allowed that will ensure the convergence of the decentralized trajectory generation for each follower.

However, this condition drastically reduces the velocity of the leader. Indeed, under this constraint the leader velocity may tend to zero for large leader-follower distances. Thus, in the next section, we study the performance of the trajectory generation algorithm when the convergence conditions are not met.

3.7.2 Leader-Follower Implementation Considerations

The condition for the convergence of the proposed algorithm was obtained in section (3.7.1). From this condition (3.52), the linear velocity of the leader robot v_{lead} has to be limited proportionally to the desired distance of the farthest follower. For formations spread over large surface areas, this limitation can be hard to satisfy (i.e. a very low velocity should be imposed on the leader robot). This condition drastically reduces the velocity and dynamics of the formation, thereby restricting the practical implementation of the leader-follower strategy for real applications. For this reason, we propose to study the dynamical behaviour of the formation when the follower velocities are saturated.

If the linear velocity of the leader robot is less than the maximal velocity of the followers, the condition (3.52) will always be satisfied if the steering angle is zero ($\xi_{lead} = 0$). On another hand for any maneuvering ($\xi \neq 0$)the desired linear velocity for a follower can be larger than its maximal velocity.

In the leader-follower strategy, the follower position is obtained by a position relative to the leader robot; thus, the desired follower velocity will be a function of the leader velocity and also of the desired relative position for the follower. As result of this condition, in some cases the imposed follower desired velocity can be greater than the follower maximal speed: it will therefore be impossible for the follower robot to track its desired position in the robot formation. For any turning manoeuvring by the leader, this condition will be evident, because the follower desired velocity will be proportional to the relative desired distance. If the leader trajectory is linear $\xi_{lead} = 0$, the desired follower velocity will be the same as the leader one, thus any difference between the leader velocity and that of the desired follower one will arise if the leader vehicle executes any turning manoeuvres. For this reason, we propose to analyze the performance of the trajectory planning algorithm using constant circular leader trajectories. By studying the trajectory generator performance for leader circular trajectories we study a critical condition, where the desired follower position need a large speed that can be higher than the follower maximal allowed speed.

If the leader robot executes a circular trajectory of radius $r_{lead} = L/\tan(\xi_{lead})$, (see Fig. 3.23), the follower will try to reach the following desired position:

$$x^d_{foll} = R^d \cos(\omega^d t); \qquad y^d_{foll} = R^d \sin(\omega^d t), \tag{3.53}$$

where:

$$\omega^d = \omega_{lead} = \frac{v_{lead} tan(\xi_{lead})}{L}$$
$$R^d = \sqrt{\frac{L^2}{\tan^2 \xi_{lead}} + \frac{l_{l-f} L \cos\gamma_{l-f}}{\tan \xi_{lead}} + l^2_{l-f}},$$

And if at each time t_k the optimal trajectory S_k is supposed to be a straight line between the follower current position $\mathbf{x}_{foll}$ and the desired one $\mathbf{x}^d_{foll}$, with velocity equal to the maximal velocity v_{max}, then, the follower position variation (eq. 3.42) can be described by:

$$\begin{aligned} \dot{x} = v\cos\theta = \\ -v_{max} \frac{x - R^d \cos(\omega^d t)}{\sqrt{\left(x - R^d\cos(\omega^d t)\right)^2 + \left(y - R^d \sin(\omega^d t)\right)^2}} \\ \dot{y} = v \sin\theta = \\ -v_{max} \frac{y - R^d \sin(\omega^d t)}{\sqrt{\left(x - R^d\cos(\omega^d t)\right)^2 + \left(y - R^d \sin(\omega^d t)\right)^2}}, \end{aligned} \tag{3.54}$$

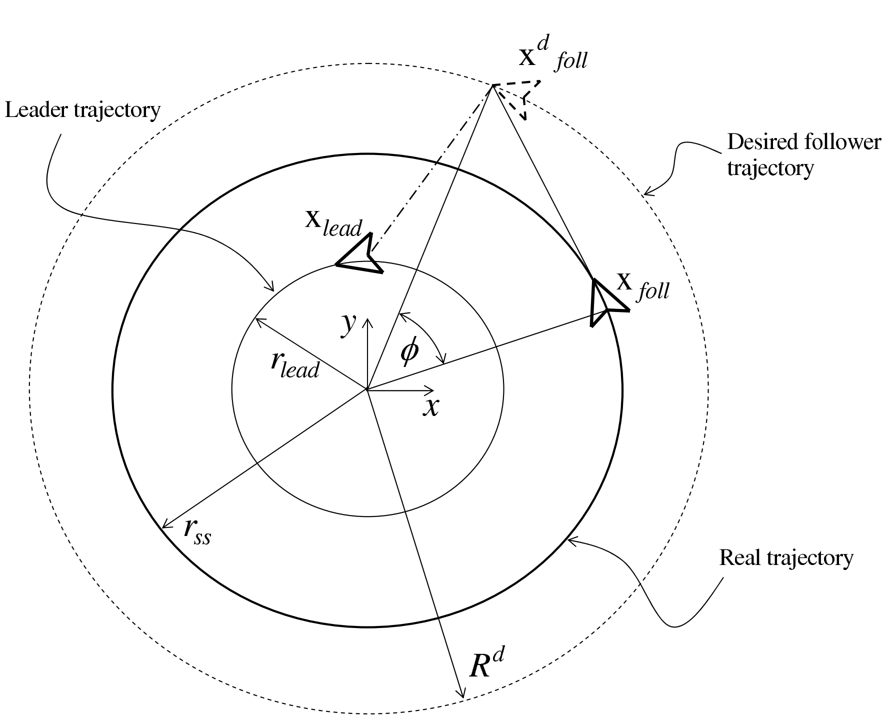


Fig. 3.23: Circular x^d_{foll} trajectory.

In order to determine the convergence of the proposed trajectory planning method, we used the Lyapunov direct method. A Lyapunov function is a scalar function $V(\mathbf{x})$ defined on a region $\mathscr{D}$ that is a continuous, positive definite $V(\mathbf{x}) > 0$, for all $\mathbf{x} \neq 0$, and has continuous first-order partial derivatives at every point of $\mathscr{D}$. The derivative of the function V with respect to the system $\dot{\mathbf{x}} = f(\mathbf{x})$, written as $\dot{V}(\mathbf{x})$, is defined as the dot product $\dot{V}(\mathbf{x}) = \bigtriangledown \dot{V}(\mathbf{x}) f(\mathbf{x})$.

The existence of a Lyapunov function for which $\dot{V}(\mathbf{x}) \leq 0$ on some region $\mathscr{D}$ containing the origin, guarantees the stability of the zero solution of $\dot{\mathbf{x}} = f(\mathbf{x}$, while the existence of a Lyapunov function for which $\dot{V}(\mathbf{x}) < 0$ is negative definite on some region $\mathscr{D}$ containing the origin guarantees the asymptotical stability of the zero solution of $\dot{\mathbf{x}} = f(\mathbf{x})$.

By defining the following variables:

$$\begin{aligned} e_1 &= x - x^d_{foll} \\ e_2 &= y - y^d_{foll} \end{aligned} \tag{3.55}$$

And its temporal derivatives:

$$\begin{aligned} \dot{e}_1 &= \dot{x} - \dot{x}^d_{foll} \\ \dot{e}_2 &= \dot{y} - \dot{y}^d_{foll} \end{aligned} \tag{3.56}$$

We can define the following Lyapunov function V:

$$V(e_1, e_2) = e_1^2 + e_2^2 \geq 0, \tag{3.57}$$

and if the follower current robot position x,y is defined by:

$$x = r_{foll}\cos(\omega_{foll}t); \qquad y = r_{foll}\sin(\omega_{foll}t). \tag{3.58}$$

The variation $\dot{V}$ can be then be written as:

$$\dot{V} = 2e_1\dot{e}_1 + 2e_1\dot{e}_1, \tag{3.59}$$

or

$$\dot{V} = 2(x - x^d_{foll})(\dot{x} - \dot{x}^d_{foll}) + 2(y - y^d_{foll})(\dot{y} - \dot{y}^d_{foll}), \tag{3.60}$$

by replacing equations (3.58 and 3.54) into equation (3.60), we obtain:

$$\begin{array}{r}\frac{dV}{dt} = 2\left(-v_{max}d + \left(r_{foll}\cos(\omega_{foll}t+\phi) - R^d\cos(\omega^d t)\right)R^d\omega^d\sin(\omega^d t)\right.\\ \left. - \left(r_{foll}\sin(\omega_{foll}t+\phi) - R^d\sin(\omega^d t)\right)R^d\omega^d\cos(\omega^d t)\right)\end{array},$$

by simplifying:

$$\frac{dV}{dt} = 2\left(-v_{max}d + r_{foll}R^d\omega^d\sin(\phi + (\omega_{foll} - \omega^d)t)\right)\ , \tag{3.61}$$

where:

$$d = \sqrt{\left(r_{foll}\cos\left(\omega_{foll}t+\phi\right) - Rd\cos\left(\omega^d t\right)\right)^2 + \left(r_{foll}\sin\left(\omega_{foll}t+\phi\right) - Rd\sin\left(\omega^d t\right)\right)^2} \tag{3.62}$$

If the system converge to a stable trajectory, then $\frac{dV}{dt} = 0$, and if R^d and R_{ss} are constant values, the frequency ω_{ss} has to be equal to ω^d. Then the variation of the Lyapunov function can be written:

$$\dot{V} = 2\left(-v_{max}d + r_{foll}R^d\omega^d\sin(\phi)\right)\ , \tag{3.63}$$

from the Figure (3.23) we can see that $\sin(\phi) = d/R^d$, then:

$$\dot{V} = 2\left(-v_{max}d + \frac{r_{foll}R^d\omega^d\, d}{R^d}\right)\ , \tag{3.64}$$

Finally if the function $\dot{V}$, is less than zero, then:

$$2\left(-v_{max}d + \frac{r_{foll}R^d\omega^d\, d}{R^d}\right) < 0 \tag{3.65}$$

the linear speed of the desired follower position is $v^d = R^d\,\omega^d$, the condition $\dot{V} < 0$ is obtained by:

$$\frac{v_{max}}{v^d} > \frac{r_{foll}}{R^d}, \tag{3.66}$$

From the condition (3.66), we can obtain the region of convergence for the follower robot. For example if $v_{max}/v^d > 1$, the region of convergence will be the region $r_{foll} <= R^d$, otherwise, for any other v_{max}/v^d rate we can obtain the region of convergence:

$$r_{foll} < R^d \frac{v_{max}}{v^d} = R_{ss}. \tag{3.67}$$

That is, for a maximal follower velocity v_{max}, and for a desired linear velocity v^d, and position x^d, y^d, the follower robot will converge to a region $r_{foll} < R_{ss}$.

3.7.3 Single Leader-Follower Example

In Figure (3.24) the convergence conditions are shown. In this simulation the leader robot drives with a constant linear velocity $v_{lead} = 5m/s$, the maximal follower speed is $v_{max} = 12.5m/s$, and a steer constant angle $\xi_{lead} = \frac{\pi}{48} rad$. The desired relative distance and angle between the leader and the follower are respectively: $l_{l-f} = 10m$ and $\gamma_{l-f} = \frac{\pi}{4} rad$, the trajectory of the follower robot is stabilized at radius $R_{ss} = 12.5m$,

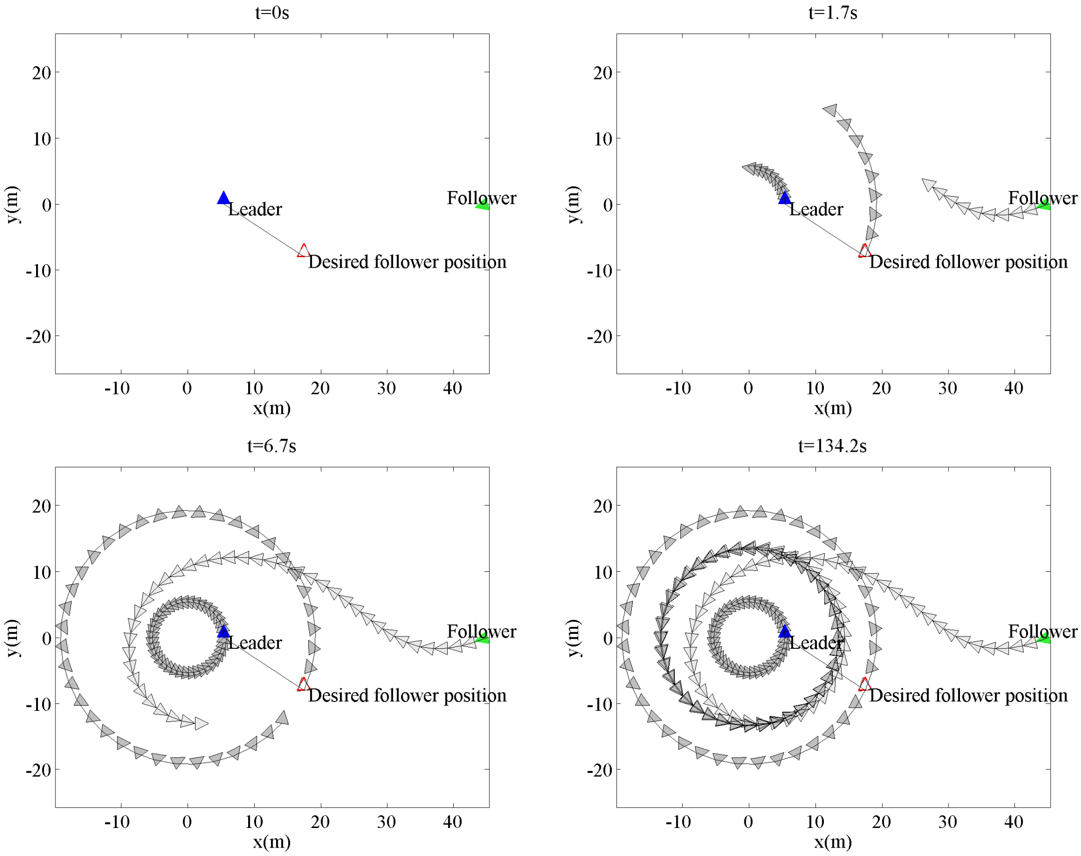


Fig. 3.24: Leader-Follower stabilization with saturated follower velocity, for a constant circular leader trajectory.$\mathcal{G} = \{1\}, \{(0,1)\}, l_{01} = 10m, \gamma_{01} = \pi/4 rad$.

The variation of relative distance l_{l-f}, and angle γ_{l-f}, are shown in the Figure 3.25. We can see the relative distance and angle variation and the steady state errors for both, distance and angle.

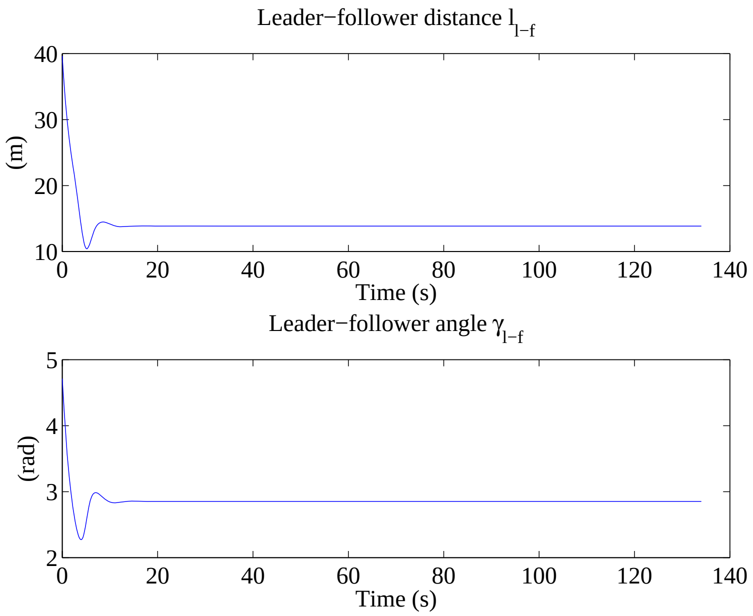


Fig. 3.25: Leader-Follower relative distance l_{l-f} and angle γ_{l-f}, for a constant circular leader trajectory

Figure 3.26 shows the performance of the proposed algorithm for a rotation maneuvering. We can see how the follower robot searches the radius R_{ss} for the rotation when the velocity is saturated. Then in the linear part of the trajectory (desired linear velocity for the follower $< v_{max}$) the follower goes through its desired position.

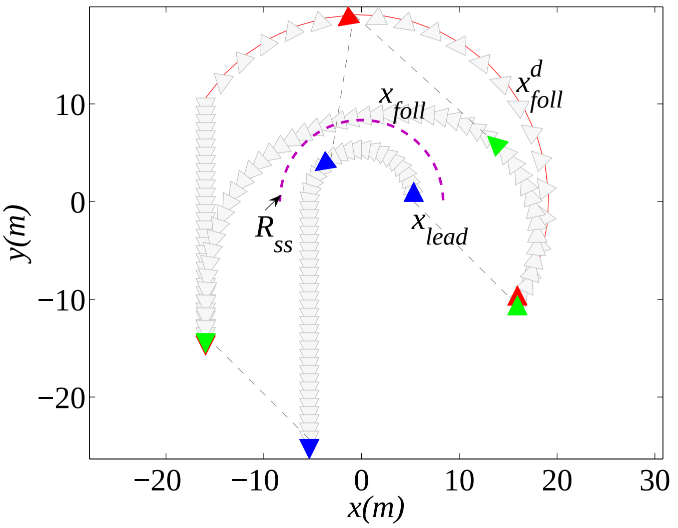


Fig. 3.26: Leader-Follower stabilization for a transient leader maneuvering with saturated follower velocity.

3.7.4 Multiple Leader-Follower Example

For a robot formation described by the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with $\mathscr{V} = \{1,2\}$ and $\mathscr{E} = \{(0,1),(1,2)\}$; and the relative desired distances and angles $l^d_{01} = l^d_{12} = 20m$, $\gamma^d_{01} = \gamma^d_{12} = \pi/4rad$. And for a leader trajectory defined by a constant linear speed $v_0 = 10m/s$ a constant steer angle $\xi_0 = \pi/90rad$, we obtain the following trajectories for a specific followers initial conditions.

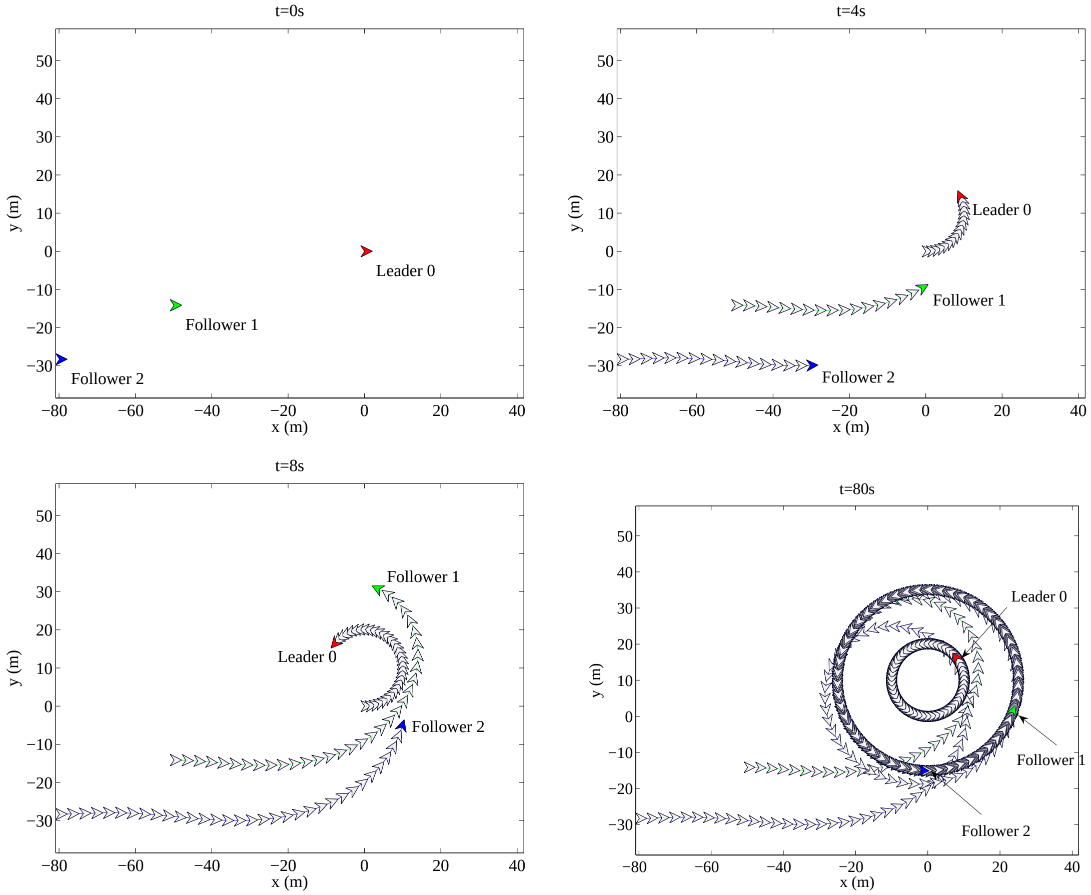


Fig. 3.27: Robot trajectories. $\mathscr{G} = \{1,2\},\{(0,1),(0,2)\}$, $l^d_{01} = l^d_{12} = 20m$, $\gamma^d_{01} = \gamma^d_{12} = \pi/4rad$, $v_0 = 5m/s$, $\xi_0 = \pi/90rad$ $v_{max1} = v_{max2} = 12.5m/s$

The followers trajectories converge to a circular path, with constant relative distance and angle l_{ij}, γ_{ij} (see Fig. 3.28). Hence, the maximal follower velocities are the same the two followers are oriented through the same region (see eq. ??).

For the same leader 0 constant trajectory we see in Figures (3.29, 3.30) the results of the trajectory planning method for a formation described by the following graph: $\mathscr{G} = \{1,2,3\},\{(0,1),(1,2),(2,3)\}$, $l^d_{01} = l^d_{12} = l^d_{23} = 20m$, $\gamma^d_{01} = \gamma^d_{12} = \pi/4rad$, $\gamma^d_{23} = \pi rad$. These figures show the same region of convergence for the three followers.

Finally we study the trajectory planner performance in a transient leader manoeuvring. When the desired follower position velocities are greater than the maximal vehicle speed, the pattern of the robot set is deformed, approaching the follower

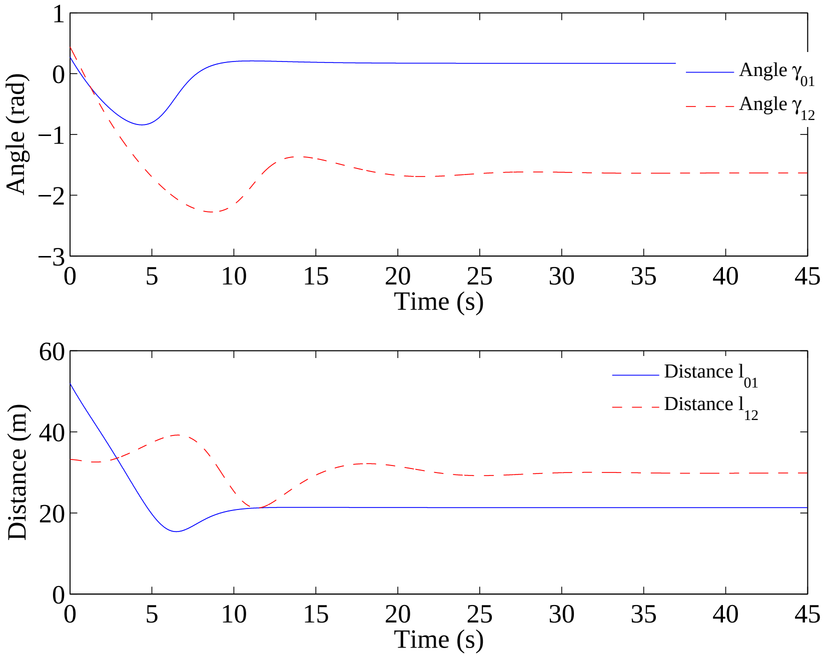


Fig. 3.28: Relative leader-follower distances and angles. Stationary leader trajectory example. $\mathscr{G} = \{1,2\},\{(0,1),(1,2)\}$, $l_{01}^d = l_{12}^d = 20m, \gamma_{01}^d = \gamma_{12}^d = \pi/4 rad$, $v_0 = 5m/s$, $\xi_0 = \pi/90 rad$, $v_{max1} = v_{max2} = 12.5m/s$

paths to an smaller radius than the desired one (see Fig. 3.31). Then, when the leader 0 trajectory is a stable path ($\xi_0 = 0$), the follower can be stabilized in the desired formation position (Fig. 3.32).

3.8 Final Remarks

We propose a new decentralized control strategy to control formations of nonholonomic robots. This approach is based on an optimal path generation method. The optimal problem was transformed into a problem of parameter optimization, and the solutions were obtained by means of a sequential feasible quadratic programming algorithm, CFSQP. The results were validated by numerical simulations and the conditions of convergence of the algorithm are presented.

In real outdoors applications, the terrain traversability is an important consideration and, therefore, any obstacle avoidance properties have to be examined. In the next section, an approach to obstacle avoidance is discussed. We include a reactive property into the proposed decentralized method.

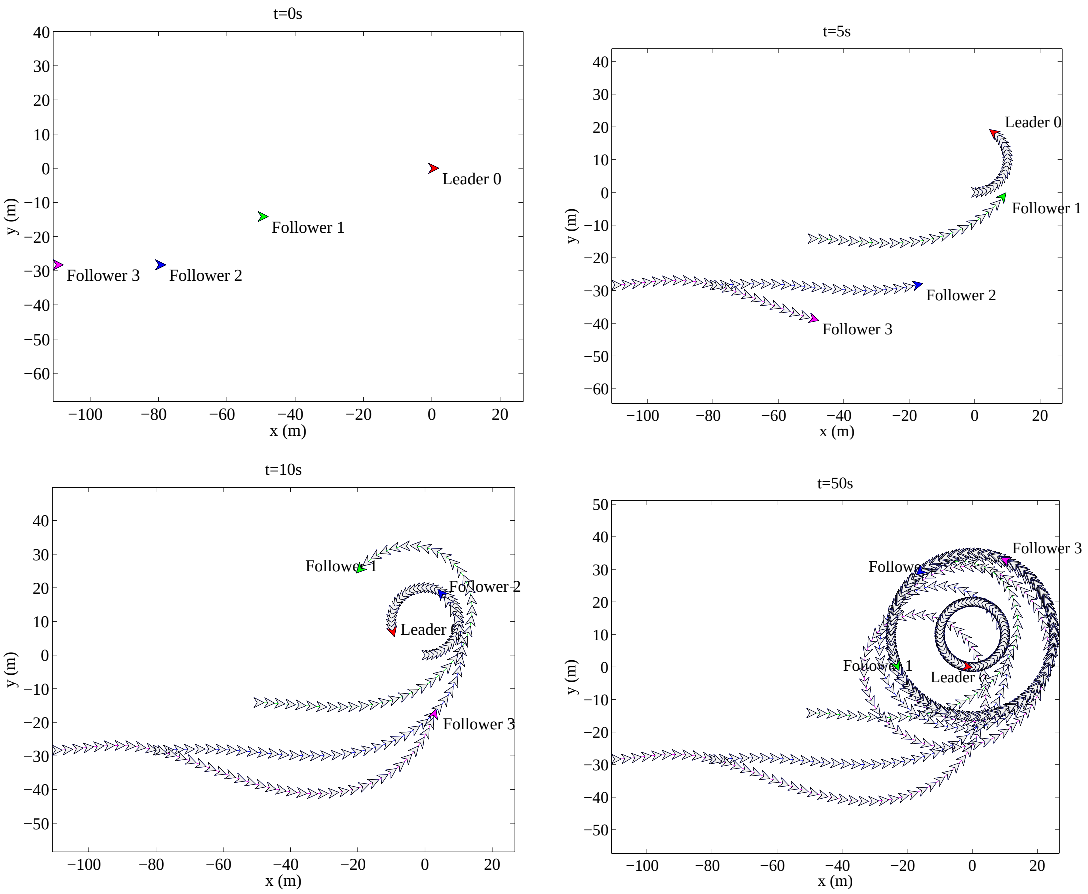


Fig. 3.29: Robot trajectories. $\mathscr{G}=\{1,2,3\},\{(0,1),(1,2),(2,3)\}$, $l^d_{01}=l^d_{12}=l^d_{23}=20m, \gamma^d_{01}=\gamma^d_{12}=\pi/4rad, \gamma^d_{23}=\pi rad$, $v_0=5m/s$, $\xi_0=\pi/90rad$, $v_{max1}=v_{max2}=v_{max3}12.5m/s$

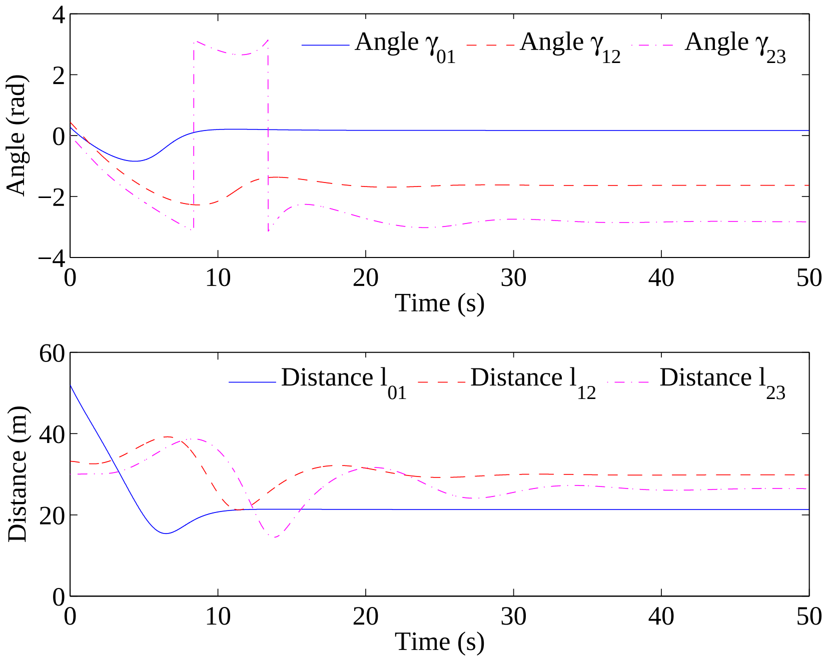


Fig. 3.30: Relative leader-follower distances and angles. Stationary leader trajectory example. $\mathscr{G} = \{1,2,3\},\{(0,1),(1,2),(2,3)\}$, $l^d_{01} = l^d_{12} = l^d_{23} = 20m, \gamma^d_{01} = \gamma^d_{12} = \pi/4rad, \gamma^d_{23} = \pi rad$, $v_0 = 5m/s$, $\xi_0 = \pi/90rad$, $v_{max1} = v_{max2} = v_{max3}12.5m/s$

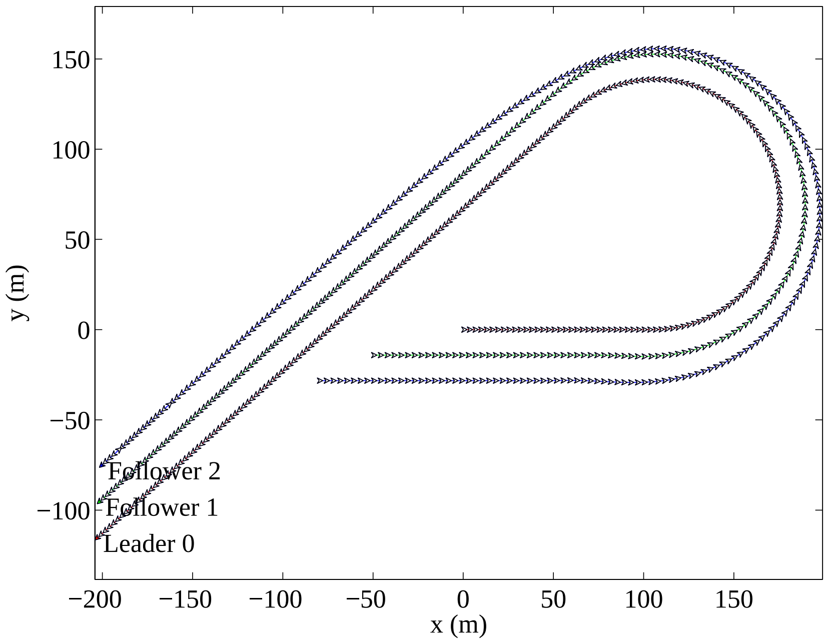


Fig. 3.31: Robot trajectories. Non stationary leader trajectory example. $\mathscr{G} = \{1,2\},\{(0,1),(1,2)\}, l^d_{01} = l^d_{12} = l^d_{23}20m, \gamma^d_{01} = \gamma^d_{12} = \pi/4rad$, $v_0 = 12.5m/s$, $v_{max1} = v_{max2} = 15m/s$

.

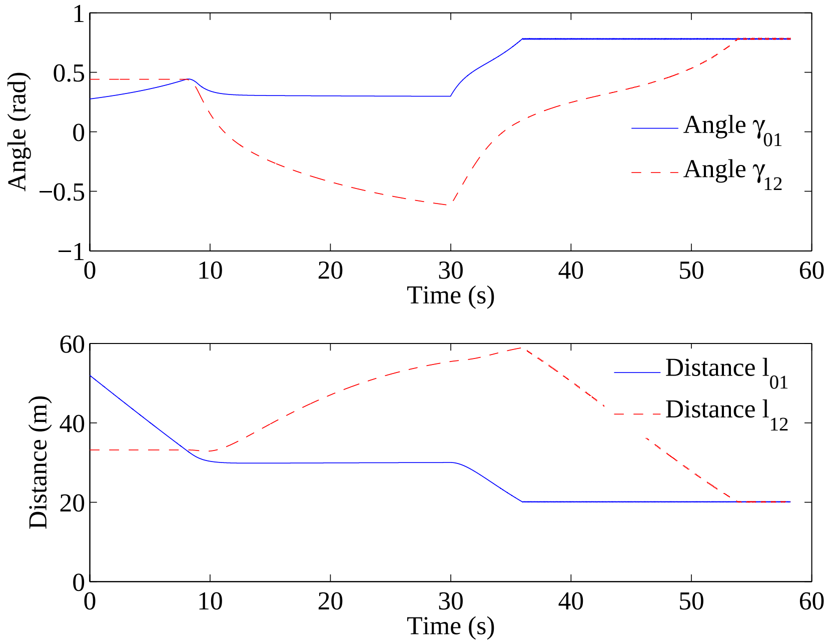


Fig. 3.32: Relative leader-Follower distances and angles. Non stationary leader trajectory example trajectory. $\mathscr{G} = \{1,2\},\{(0,1),(1,2)\}$,$l^d_{01} = l^d_{12} = l^d_{23}20m, \gamma^d_{01} = \gamma^d_{12} = \pi/4rad$, $v_0 = 12.5m/s$, $v_{max1} = v_{max2} = 15m/s$.

4

Obstacle Avoidance

4.1 Introduction

This chapter proposes an approach to the problem of obstacle avoidance for robot formations. Typically, obstacle avoidance strategies in multi-robot systems are based on hierarchical hybrid systems (i.e. [45]). Under this approaches when the robot detects an object that is closer than a threshold value, an obstacle avoidance process is used. Otherwise, if the robot does not detect any obstacle any mission process is used. Therefore, the stabilization strategy is composed of different control states.

In this work, the obstacle avoidance strategy is incorporated into the global control strategy. Thus, different control states are not required. We define a reactive term that is included as an optimization criterion in the minimization problem (3.28). This reactive term is inspired by the Deformable Virtual Zone (DVZ) method ([4]).

4.2 Deformable Virtual Zone

This section presents an obstacle avoidance algorithm based on the use of a continuous Deformable Virtual Zone (DVZ). The main idea is to define the robot/environment interaction as a DVZ surrounding the vehicle (Fig. 4.1). The deformation of this risk zone Ξ is due to the intrusion of proximity information and thus controls the robot reactions. This DVZ characterizes the deformable zone geometry and depends on the robot velocities (forward and rotational velocities, v and ω). Briefly, the risk zone, when disturbed by obstacle intrusion, can be reformed by acting on the robot velocities.

The undeformed DVZ is defined as an elliptic shape around the robot and is aligned with the robot orientation (see Fig. 4.2). The analytical expression of the polar signature of the undeformed DVZ, in the robot frame B, can be expressed as:

$$d(\alpha) = \frac{-B + \sqrt{B^2 - 4AC}}{2A}, \tag{4.1}$$

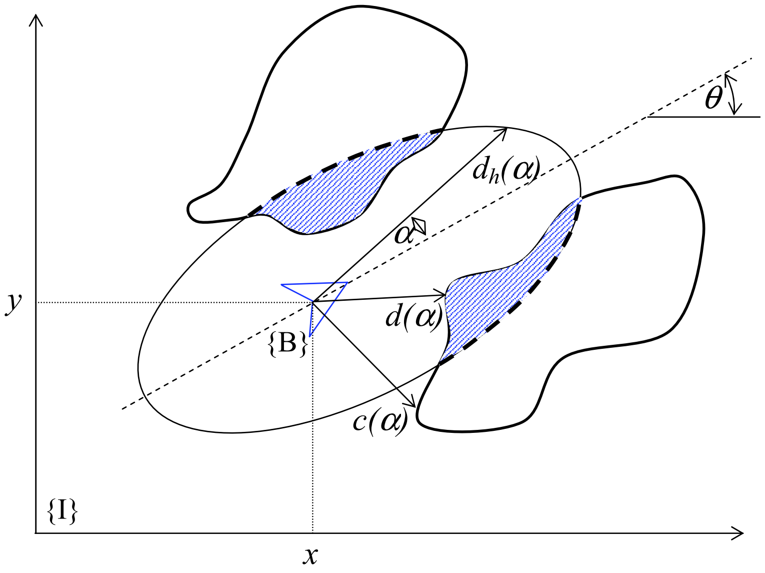


Fig. 4.1: Deformable virtual zone definition

with:

$$\begin{aligned}
A &= (c_1 \cos\alpha)^2 + (c_2 \sin\alpha)^2 \\
B &= 2\left(a_1 \cos\alpha c_2^2 + a_2 \sin\alpha c_1^2\right) \quad , \\
C &= (a_1 c_2)^2 + (a_2 c_1)^2 - c_1 c_2
\end{aligned} \tag{4.2}$$

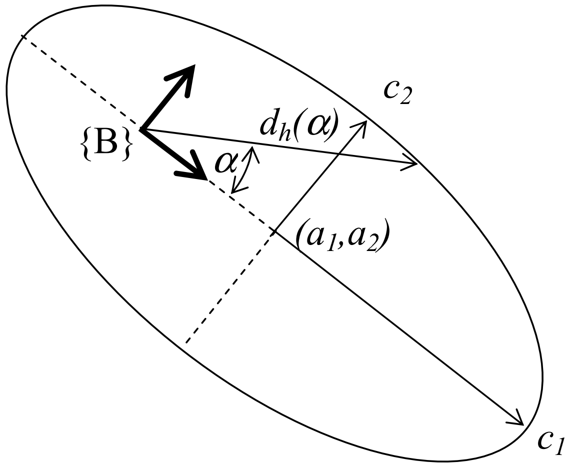


Fig. 4.2: Undeformed DVZ

The undeformed DVZ is function of the robot linear velocity v; hence, we choose the DVZ parameters as:

$$\begin{aligned}
c_1 &= c_1^{min} + (1-r)\, v(t) \\
c_2 &= c_2^{min} + (1-r)\,(1-p)\, v(t) \quad , \\
a_1 &= sign\,(v(t))\, c_1 \\
a_2 &= 0
\end{aligned} \tag{4.3}$$

where, $0 < p, r < 1$ are arbitrary coefficients, and c_1^{min}, c_2^{min} the desired minimal ellipse axes.

The deformed virtual zone $d(\alpha)$ is acquired from the sensor information, denoted $c(\alpha)$. Since the meaningful information is restricted to that inside the undeformed DVZ, a preliminary test on the sensor information is necessary:

$$\begin{array}{rcl} d(\alpha) & = & d_h(\alpha)(1 - I\cos(\alpha)) \qquad \textbf{if} \qquad c(\alpha) < \Theta \\ & = & d_h(\alpha) \qquad\qquad\qquad \textbf{elsewhere} \end{array}, \tag{4.4}$$

where, Θ is the safe range zone, and $I = \Theta - c(\alpha)$ the measured intrusion.

From equation (4.4) we can see that the DVZ is subject to a deformation $d_h(\alpha) I \cos(\alpha)$ in the sensor direction α. This deformation is greater in the vehicle's front direction $\alpha = 0$; also the deformation is proportional to the vehicle's linear velocity.

The original DVZ approach [4], searches the robot's control (linear velocity and steer angle) that minimizes the variation in DVZ deformation. In a similar vein, we propose to minimize the DVZ deformation. However, this minimization is performed in a horizon of time; that is, we searched for a trajectory that would also minimize the deformations in the DVZ. In the next section, this proposition is developed and discussed.

4.3 Deformable Virtual Zone in the Trajectory Generator

In chapter 3, we proposed a decentralized trajectory planner for nonholonomic robot formations. The reference trajectory for each robot was obtained by minimizing a predefined performance index. We then obtained the solution for a desired minimal time trajectory for each robot. In this section, we propose a composite performance index that includes a self-called reactive term based on the DVZ method.

In simple terms, the proposed reactive term penalizes the DVZ deformation over the planned trajectory. This reactive optimization term for the instant t is defined as:

$$J_{obst}(t) = \int_{\alpha_i(t)}^{\alpha_f(t)} (d_h(\alpha) - d(\alpha))\, d\alpha \tag{4.5}$$

That is, the $J_{obst}(t)$ computes the total deformation of the DVZ at the instant time t, where α_i, α_f are the initial and final sensor angles (see Fig. 4.3).

This reactive term $J_{obst}(t)$ is included in a composite performance index j. With this, the optimization problem is proposed as:

$$J = \int_0^{t_f} (\alpha_t t + \alpha_{obs} J_{obst}(t))\, dt, \tag{4.6}$$

where, $0 < \alpha_t, \alpha_{obst} < 1$ are the composite performance index's weights. With the initial and trajectories constraints (eq. 3.13 and 3.14).

We were interested in applications where the robot measures the distance by means of ultrasonic sensors. The proposed experimental robots are equipped with

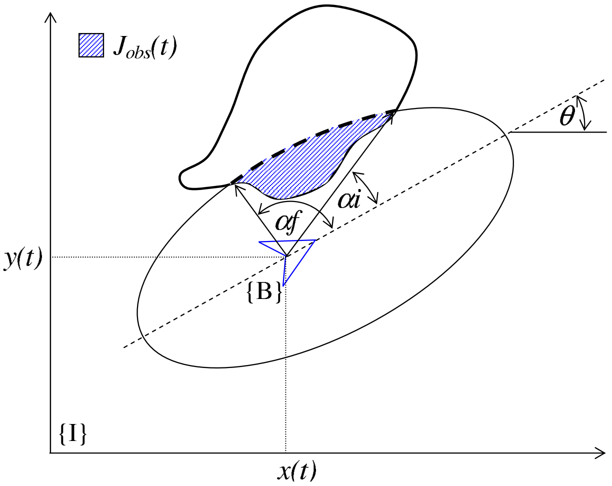


Fig. 4.3: Total DVZ deformation $J_{obst}(t)$

four ultrasonic sensors in predefined directions. Therefore, the DVZ is not continuous and do not covers all of directions around the vehicle; only the sensor directions are taken into account to compute it. In this case, we obtain the DVZ deformation for these predefined directions α_m (see Fig. 4.4):

$$J_{obst}(t) = \sum_{m=1}^{M} (d_h(\alpha_m) - d(\alpha_m)) \tag{4.7}$$

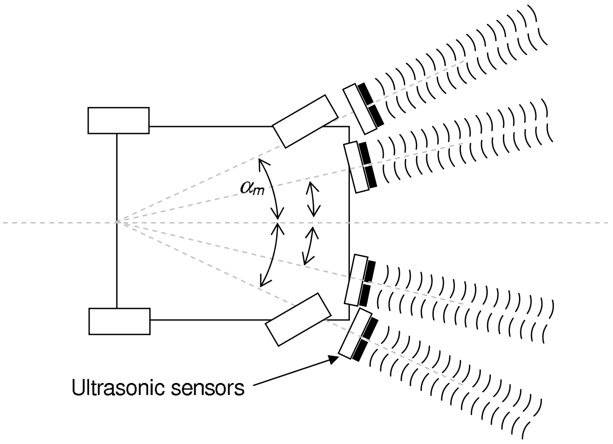


Fig. 4.4: Robot's ultrasonic sensors

4.3.1 DVZ in the Trajectory Generator

At each sampling instant t_k, we use the ultrasonic sensor information to minimize the proposed performance index over the time horizon $[t_k, t_f]$. Then for the parameterized trajectory 2, the DVZ deformation $d(\alpha) - d_h(\alpha)$ is computed for detected obstacles at time $t = t_0$ for the whole time interval $[t_0, t_f]$ (see Fig. 4.5). For the next sampling time t_{k+1}, the obstacle information is then actualized.

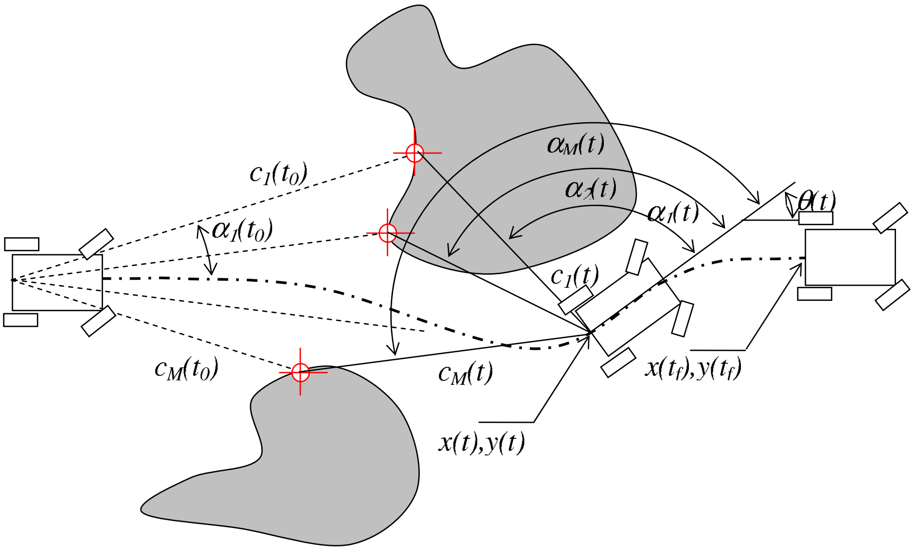


Fig. 4.5: J_{obst} computing over the planning trajectory for the time interval $[t_0, t_k]$

The optimal problem with the composite performance index for the time and DVZ deformation minimization can be written as:

$$J = \int_{t_0}^{t_f} \alpha_t t + \alpha_{obs} J_{obst}(t) dt. \tag{4.8}$$

this composite performance index (4.8) can be transformed in the following parameters minimization problem:

$$\min_{\mathbf{C},t} G(\mathbf{C}, t), \tag{4.9}$$

with

$$G = \alpha_t t_f + \alpha_{obst} \sum_{p=1}^{P} \mu_p J_{obst}(\tau_p, C), \tag{4.10}$$

$\tau_p \in [0, 1]$, μ_p, the quadrature coefficients, and P the number of points for the quadrature aproximation.

The undeformed DVZ $d(\alpha)$ coefficients (eq. 4.3), are computed with the velocity in the normalized time τ space:

$$v(\tau) = \frac{\sqrt{\dot{z}_1(\tau)^2 + \dot{z}_2(\tau)^2}}{t_f}.$$

4.3.1.1 Simulations

Given a formation of three robots, $\mathscr{V} = \{2, 3\}$ with the leader robot 1 and the desired relative positions: $l_{12} = l_{13} = 10m, \gamma_{12} = -\gamma_{13} = \frac{\pi}{4}$. We applied the trajectory generation method with the composite performance index (4.8).

In Figure (4.6) we see the robot formation $\mathscr{V}$ in an obstacle-filled environment. We assume that each robot is equipped with four ultrasonic sensors (see Fig. 4.4), where the sensor angles are: $\alpha_1 = -3\pi/4, \alpha_2 = -\pi/6, \alpha_3 = 3\pi/4, \alpha_4 = \pi/6$. We also assume that these sensors can detect any obstacle within a range of 10 meters.

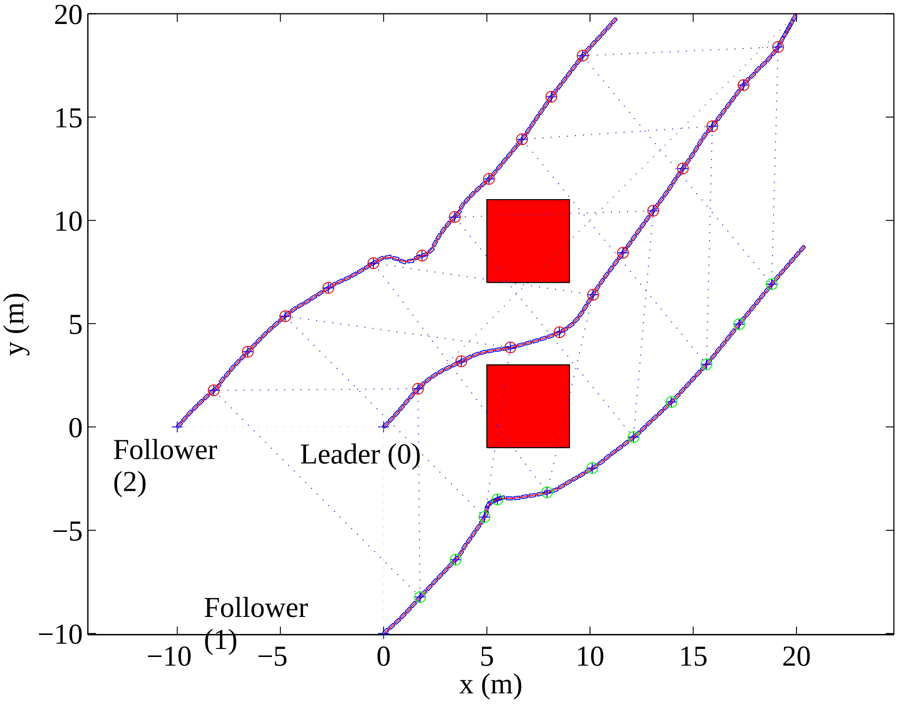


Fig. 4.6: Robot's trajectories. Formation: $\mathscr{V} = \{1,2\}$ with the leader robot 0. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Bounded computing time $T_s = 300ms$

The robot trajectories are shown in Figure (4.6). To avoid obstacles, the formation's geometrical pattern is deformed and returns to the desired pattern once there are no more obstacles.

4.4 Real-time Considerations

In section (3.6), we presented a nonlinear trajectory generator. This trajectory generator is based on real-time path generation. As the algorithm of optimization, we proposed the feasible nonlinear programming solver CFSQP [78]. An ideal solution in critical real-time trajectory problem is to obtain a feasible solution a each iteration, obtaining a feasible solution is better than none at all.

In this section, we show the simulations of trajectory generation to study the computing time. We present the trajectory generation method for a single robot. This robot has to displace from an initial state to a final one while avoiding obstacles. The aim of these simulations is to contrast the solutions for different computing time restrictions in order to validate the results of the feasible nonlinear programming solver in bounded computing time conditions.

In the first simulation, we see the trajectory when there are no computing time restrictions (Fig. 4.8-b) In this simulation, the computing time of each partial trajectory is variable, and it depends on the solver convergence conditions. Also,

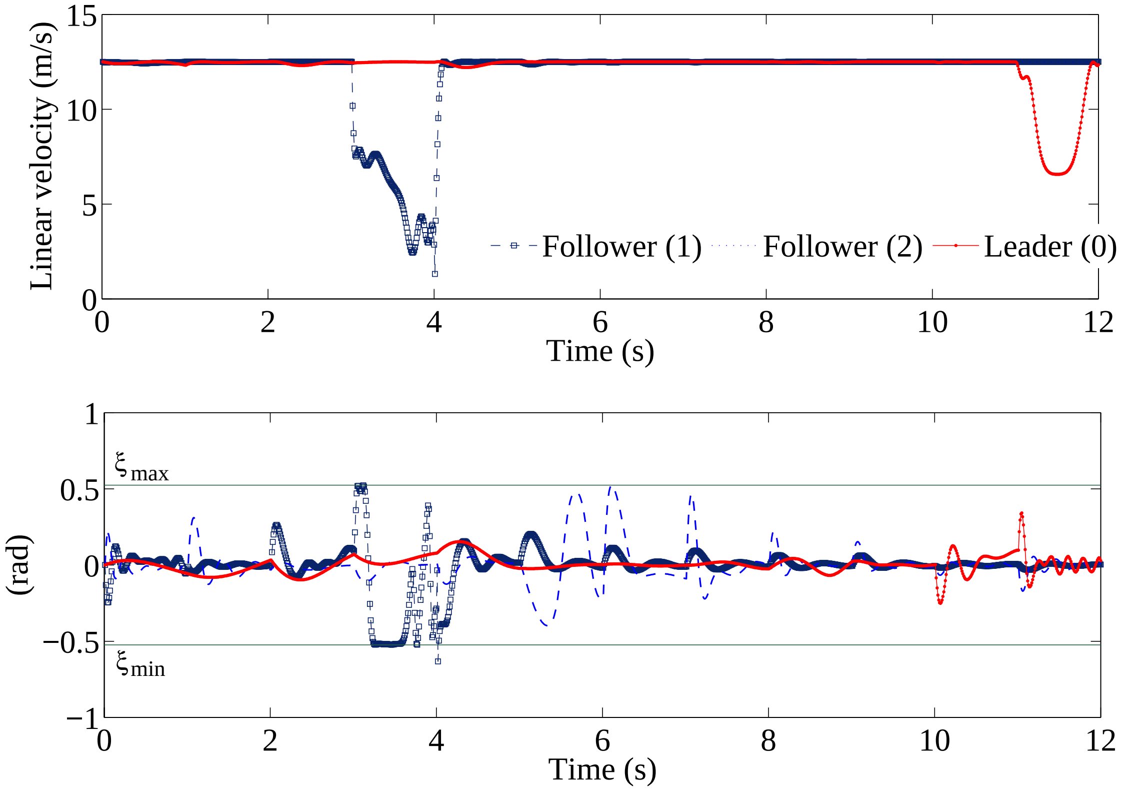
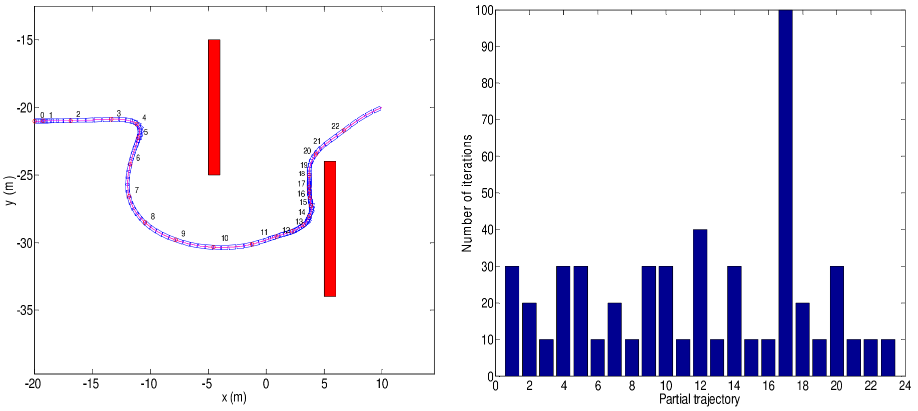


Fig. 4.7: Up: robots velocities, down: robot's steering angle. Formation: $\mathscr{V} = \{1,2\}$ with the leader robot 1. Nodes $\mathscr{E} = \{(0,1),(0,2)\}$. Desired relative positions $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Bounded computing time $T_s = 300ms$

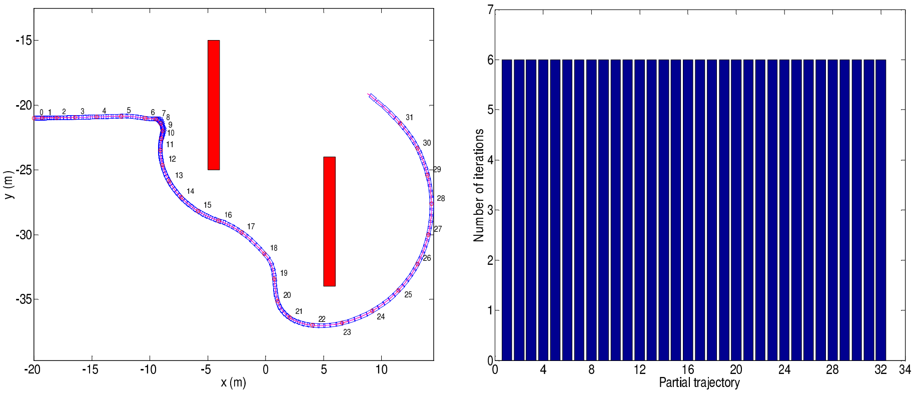
Figure 4.8-b) shows that the maximal computing time is achieved when the obstacles are sensed; in this case, the composite performance index is more complex and more computing time is therefore required.



(a) (b)

Fig. 4.8: Trajectory generation simulation: unbounded time condition. (a) Trajectory. (b) Iterations number

In real applications a, there is a computing or possessing time constraint. In the next figure (4.9) we see the simulation of a single vehicle, but with bounded



(a) (b)

Fig. 4.9: Trajectory generation simulation: bounded time condition. (a) Trajectory. (b) Iterations number

computing time. We can see that this trajectory differs from the previous one (4.8): a longer trajectory is obtained from the bounded time condition than from the unbounded condition.

If the time is bounded, we can obtain solutions that do not satisfy the convergence conditions for the nonlinear programming solver, but these solutions are feasible and can be used as reference paths.

4.5 Final Remarks

In this section we adapted the DVZ method to be included in the trajectory generator. The original DVZ method minimizes the virtual zone deformation by choosing the controls of the robot in this work we proposed minimize the virtual zone deformation over a feasible path. Thus, the reference trajectory is modified to reduce the virtual zone modifications.

Many optimal path planning method often deals with the obstacle avoidance problem by including the obstacle as a trajectory inequality constraint e.g. [80]. For example, if the obstacle is represented as a circumference, the obstacle region can be represented as an inequality constrained region in the x, y plan. This constraint is then included in the optimization problem to obtain a free collision path. In contrast to these approaches, the obstacle avoidance by DVZ can be applied for unknown geometrical obstacles patterns. Additionally, the minimization of the DVZ is obtained not only changing the robot orientation but also reducing the robot velocity; hence, in the proximity of obstacles the robot speed is modified to minimizes the DVZ deformation; this is a secure way to traverse obstacle environments. Finally, by including the DVZ in the trajectory generator, the minimization is obtained taking into account the control constraint (control saturation).

5

Wireless Communication for Relative Positioning in Multi Mobile Robots Formation

5.1 Introduction

In cooperative robotic systems with explicit communication, radio or wireless networks are widely used for data transmission between robots. The properties of these networks, such as signal propagation phenomena, suggest their utility as positioning instruments, i.e. the radio received signal strength (RSS) can be exploited to estimate the positions between robots. This means that the communication structure can be used when conventional positioning systems like GPS are unavailable [81].

GPS is an extended method for outdoor location in robotics applications. The GPS method is based on frequencies that require a line-of-sight between the sensor and satellites to transmit signals. If no line-of-sight is available, as is the case indoors or in other foiled situations like urban areas, the GPS location will be inaccessible or an erroneous measurement will be deployed by the sensor. The use of the RSS from wireless networks is a field of study with applications in environmental monitoring, structural monitoring, military battlegrounds and public safety [82], e.g. the United States requires cellular operators to estimate the position of an emergency caller with an error of less than 100 m [83]. Positioning by signal strength is supported by the IEEE 802.15.4 standard and will be the key feature for the ongoing IEEE 802.15.4a revision project [84]. In the robotic field the RSS of WiFi networks has been proposed to positioning robotic systems in indoor applications[85]. The conventional techniques for localization by using radio signals are based on statistical models, i.e. could be considered the variation of the RSS individual measurement around the mean value has a normal distribution in dB [86–88].

In this chapter, we propose to fuse WiFi signal measurements with vehicle sensor values. We then estimate the relative positioning between vehicles using this fused signal. A popular data fusion method is the Extended Kalman Filter (EKF), a variation of the Kalman filter). The EKF is a variation of the Kalman filter [89] to solve nonlinear problems. Other variations of the Kalman filter has been used as data fusion method for vehicles positioning like the Unscented Kalman filter (UKF) [90],A comparative study of the EKF and UKF in positioning problems was reported in [91]. In [92] a Kalman filter approach is used in multiple robots localisation. In vehicles positioning problem the EKF has been used to integrate the differential global positioning system (DGPS) signals and the vehicle sensors [93]. In this work, we use the EKF to estimate the relative positions between vehicles by fusing the WiFi network and the vehicle sensors. The EKF is based on the kinematical nonlinear model of a four-wheel car-like vehicle. Each vehicle measures the strength and direction of the maximal reception of the wireless signal received from its nearest neighbour, while measuring its wheel speed and steering angle. By fusing these signals, the robot is able to estimate its relative position with respect to its neighbour

5.2 RSS-Based Location Estimation

The problem of RSS-based location estimation can be defined as the following. Suppose a leader target sensor node is located at some unknown location (x_i,y_i) and the follower reference sensor node located at known location (x_j,y_j), we want to estimate the unknown location coordinates of the target node by measuring the RSS of Wifi signals transmitted by the reference vehicle node sensor. The location of the leader robot can be obtained from the leader-follower distance l_{ij} and relative angle γ_{ij} by solving the following non linear equation system, $\forall (i,j) \in \mathscr{E}$:

$$\begin{array}{lcl} \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} & = & l_{ij} \\ \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) & = & \gamma_{ij} \end{array} \tag{5.1}$$

5.2.1 Propagation Phenomena

Propagation mechanisms are very complex and diverse. First, because of the separation between transmitter and receiver, signal strength may be attenuated. In addition, the signal propagates by means of diffraction, scattering, reflection, transmission, refraction, etc.

Diffraction occurs when the direct line-of-sight propagation between the transmitter and receiver is obstructed by an opaque obstacle whose dimensions are considerably larger than the signal wavelength. Diffraction occurs at the obstacle edges where the radio waves are scattered and, as a result, are further attenuated. The diffraction mechanism allows radio signals to be received when the line-of-sight conditions are not satisfied (no-line-of-sight), in either urban or rural environments.

Scattering occurs when the propagation path contains the obstacles whose dimensions are comparable to the wavelength. This phenomenon is similar to the diffraction, except that the radio waves are scattered in a greater number of directions. Of all the mentioned effects, scattering is the most difficult to be predicted.

Reflection occurs when the radio wave impinges an obstacle whose dimensions are considerably larger than those of the incident wave. A reflected wave can either decrease or increase the signal strength at the reception point. In cases of many reflected waves, the received signal level tends to be very unstable. This phenomenon is commonly referred to as multipath fading, and the signal is often Rayleigh distributed.

Transmission occurs when a radio wave encounters an obstacle that is to some extent transparent for the wave. This mechanism allows the reception of radio signals inside buildings in cases where the actual transmitter location is either outdoors or indoors.

Refraction is very important in macrocell radio system design. Due to the inconstancy of the atmospheric refractive index, radio waves do not propagate along a straight line, but rather along a curved one. Therefore, the coverage area of an actual transmitter is usually larger. However, as a result of the fluctuations in atmospheric parameters, the received signal strength fluctuates as well.

Since there is frequently no line-of-sight [ok like this here] between the transmitter and receiver, the received signal is a sum of components that often stem from several of the above-described phenomena.

There are thus two major sources of error in the measurement of location metrics: multipath fading and no-line-of-sight [but here this is correct] conditions due to shadow fading [94] (see Fig. 5.1). The IEEE 802.11b standard uses the industrial, scientific and medical (ISM) radio frequency (2.4-GHz), which is license-free around the world. Accurate prediction of signal strength from location is a complex and difficult task, since the signal propagates by unpredictable means [95]. In the 2.4-GHz frequency band, microwave ovens, Bluetooth devices, 2.4-GHz cordless phones, and welding equipment can be sources of interference. Signals with this frequency are absorbed by water and, consequently, people will also absorb signal since human bodies are almost 70% water. Due to reflection, refraction, scattering, dependence on atmospheric parameters, and absorption of radio waves by structures inside a building, the transmitted signal most often reaches the receiver by more than one path, resulting in a phenomenon known as multipath fading [96]. Signal multipath effects cause the observed signal strength to vary in unpredictable ways as the receiver position varies, but signal profiles tend to remain approximately the same over short distances [96].

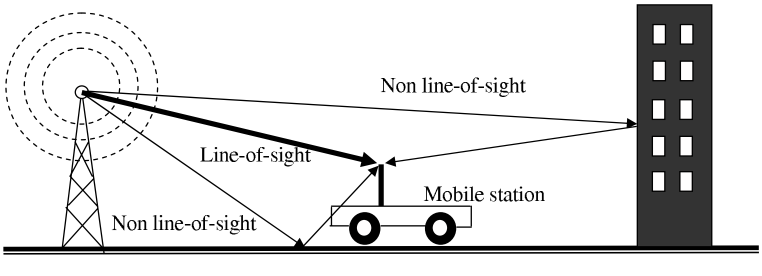


Fig. 5.1: Multipath effects on a mobile station

Each subsystem or follower robot j is able to measure the RSS of the wireless communication channel with its leader i. The RSS can be related to the distance through the narrowband radio propagation path-loss model [82]:

$$R_p = R_0 + 10 \cdot \alpha_{dbm} \cdot \log_{10}(l) + \nu, \tag{5.2}$$

where R_0 is the signal power loss in dBm at a distance of 1 metre, R_p is the signal loss in dBm at a distance of $l > 1$ meter, α_{dbm} is the exponential path-loss coefficient, and ν is a Gaussian variable representing lognormal shadow fading effects in multipath environments. The random variable ν can be considered a zero-mean gaussian variable $\nu \sim N(0, \sigma_n^2)$, with a standard variation σ_n that depends on the characteristics of the multipath environment [97]. If the transmission power R_t of the WiFi network is known, we can determine R_p at the robot j by: $R_p = R_t - R_r$, where R_r is the measured power at the reception node (robot j) [66].

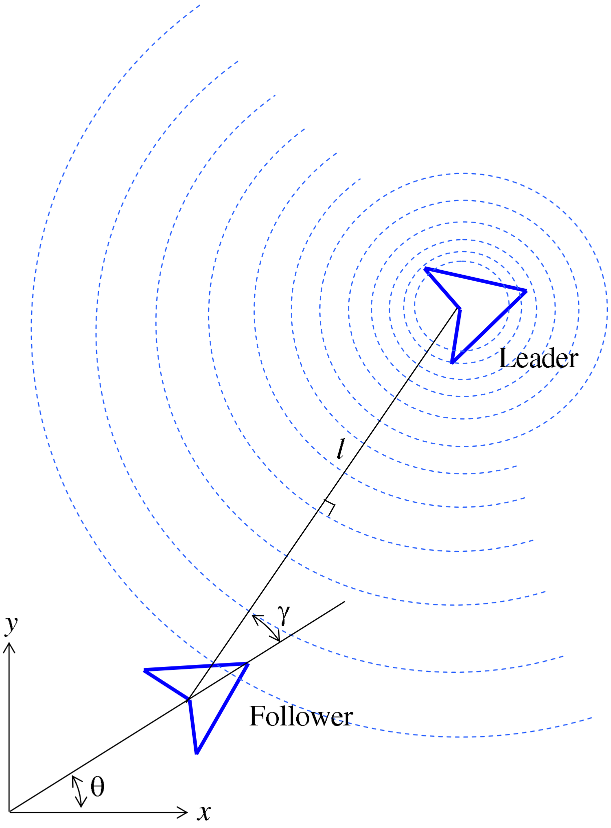


Fig. 5.2: RSS location estimation

In order to determine at node j the direction γ_{ij} of the source (node i) of the WiFi signal, we assume that each robot is equipped with a directional antenna or steerable directional antenna [98]. Being directional, this antenna receives signals over a very narrow bandwidth. Hence, a maximum value for the RSS R_r is reached when the antenna is pointed directly at source j. The direction γ_{ij} can be expressed as:

$$\tilde{\gamma}_{ij} = \gamma_{ij} - \phi_{ij} + \rho, \tag{5.3}$$

ϕ_{ij} is an angle difference due to the no-line-of-sight environments,[87], ρ is the noise in the measurement, and γ_{ij} is the line-of-sight angle between the source i and the node j.

5.2.2 RSS Propagation Model Identification

In order to obtain the RSS propagation model coefficients R_0, α_{dbm} from equation (5.2), we measured the RSS from the robot for different distances. The results are shown in Figure (5.4).

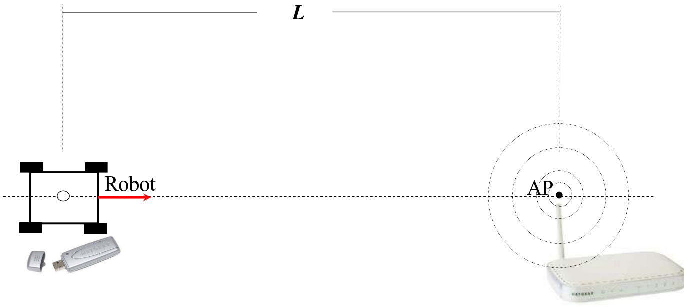


Fig. 5.3: Identification of RSS propagation model. Experimental setup

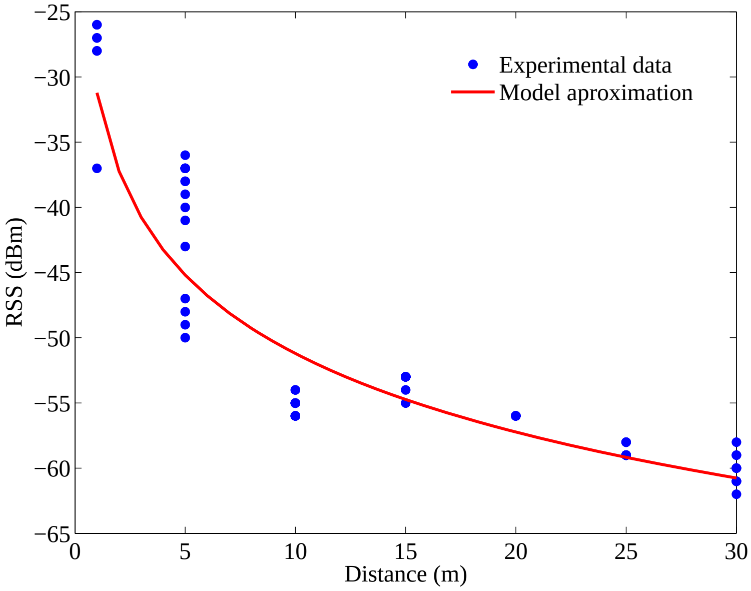


Fig. 5.4: Results for identification of RSS path-loss model

From the experimental results we can estimate the model coefficients R_0, α_{dbm} (5.2) by using least squares minimization. Thus the model can be written as:

$$R_p = -28.0055 + 10 \cdot -2.225 \cdot \log_{10}(l) + \nu. \tag{5.4}$$

By using the identified path-loss model 5.4 we can estimate the distance between two formation robot nodes (i, j). However, to use relation 5.4, we need to filter the noise ν. In the next section, we propose to fuse the R_p and odometric vehicle measurements via an extended Kalman filter for estimating the robot positioning in the formation.

5.3 Position Estimation via Extended Kalman Filter

The measurement equation for the system (3.6) can be written by using the measurement equations (5.2-5.9):

$$\mathbf{y}_j = \mathbf{h}_j(\mathbf{x}_j, t) + \mathbf{v}_j(t), \tag{5.5}$$

with $\mathbf{y}_j = \left[R_r \, \tilde{\gamma}_{ij} \, \tilde{\theta}_j \, \tilde{\xi}_j \, \tilde{v}_j \, \tilde{\omega}_j\right]^T$, being R_r the measurement of the RSS of the wireless network, $\tilde{\gamma}_{ij}$ the angle of arrival of the wireless signal and $(\tilde{\theta}_j, \tilde{\xi}_j, \tilde{v}_j, \tilde{\omega}_j)$, the orientation and velocities measurements. The vector $\mathbf{v}_j$ is the measurement noise vector.

Each robot j has an odometric speed v_j sensor, an electronic compass for orientation θ_j measurement, and a potentiometer as the steering angle ξ_j sensor. The measurements can be modeled as:

$$\tilde{v}_i = v_i + \eta, \tag{5.6}$$

$$\tilde{\omega}_i = \omega_i + \lambda, \tag{5.7}$$

$$\tilde{\theta}_i = \theta_i + \iota, \tag{5.8}$$

$$\tilde{\xi}_i = \xi_i + \kappa, \tag{5.9}$$

where η, λ ι and κ are, respectively, the noise in the linear speed, the steering angular velocity, and the orientation and steer angle sensors.

For a sampling time T_s, the Zero-Order-Holder (ZOH) discrete approximation of the model 3.7, and measurements 5.5, can be expressed at time t_{k+1}, as:

$$\begin{aligned} \mathbf{r}_{ij}(t_{k+1}) = \mathbf{r}_{ij}(t_k) + [\mathbf{f}_j(\mathbf{r}_{ij}(t_k), \mathbf{u}_j(t_k)) + \ldots \\ \tilde{\mathbf{f}}_j(\mathbf{r}_{ij}(t_k), \mathbf{r}_{ij}(t_k), \mathbf{u}_i(t_k))] \cdot T_s + \mathbf{w}_j(t_k), \end{aligned} \tag{5.10}$$

$$\mathbf{y}_j(t_k) = \mathbf{h}_j(\mathbf{r}_{ij}(t_k)) + \mathbf{v}_j(t_k), \tag{5.11}$$

The process noise $\mathbf{w}_j$ and the measurement noise $\mathbf{v}_j$ are assumed to be zero-mean, white noise with covariance properties as follows:

$$E[\mathbf{w}(k)\mathbf{w}^T(j)] = \begin{cases} Q(k), & k = j \\ \mathbf{0}, & k \neq j \end{cases}, \tag{5.12}$$

$$E[\mathbf{v}(k)\mathbf{v}^T(j)] = \begin{cases} R(k), & k = j \\ \mathbf{0}, & k \neq j \end{cases}, \tag{5.13}$$

$$E[\mathbf{w}(k)\mathbf{v}^T(j)] = \quad \mathbf{0}, \tag{5.14}$$

for all k and j.

The EKF is based on two main steps. A *time update* projects the current state estimate in time, and a *measurement update* adjusts the projected estimated by an actual measurement at each time [68]. For the linear approximation of the process and measurement models(5.10) and (5.11), the EKF is based on two main sets, the time update and the measurement update. These sets are defined by:

Time update set:

$$\hat{\mathbf{x}}_j^-(t_{k+1}) = \hat{\mathbf{r}}_{ij}(t_k) + [\mathbf{f}_j(\hat{\mathbf{r}}_{ij}(t_k), \mathbf{u}_j(t_k)) + ... \\ \tilde{\mathbf{f}}_j(\hat{\mathbf{r}}_{ij}(t_k), \hat{\mathbf{r}}_{ij}(t_k), \mathbf{u}_i(t_k))] \cdot T_s, \tag{5.15}$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1}, \tag{5.16}$$

and the *measurement update set*:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}, \tag{5.17}$$

$$\hat{\mathbf{x}}_j(t_k) = \hat{\mathbf{x}}_j(t_k)^- + K_k \left(\mathbf{y}_j(t_k) - \mathbf{h}_j\left(\hat{\mathbf{x}}_j^-(t_k)\right)\right), \tag{5.18}$$

$$P_k = (I - K_k H_k) P_k^-, \tag{5.19}$$

being the jacobian:

$$A_{k[m,n]} = \frac{\partial\left(f_{j[m]} + \tilde{f}_{j[m]}\right)}{\partial x_{j[n]}}(\hat{\mathbf{x}}_j(t_{k-1}), \mathbf{u}_j(t_k), \mathbf{u}_i(t_k))),$$

and,

$$H_{k[m,n]} = \frac{\partial h_{j[m]}}{\partial x_{j[n]}}(\hat{\mathbf{x}}_j^-(t_k)),$$

where P_k is the error covariance and K_k the Kalman filter gain matrix. The discrete EKF algorithm (equations (5.15) to (5.19)) is a recursive process. As such it requires initialization prior to starting the recursion. If we assume that the first measurement occurs at t_1, the initialized state estimate and error covariance $\hat{\mathbf{x}}_j(t_0)$ and P_0 should be given. The EKF is tuned by choosing the noise covariance matrixes Q_k and R_k.

5.3.1 Simulations

To validate the proposed EKF we first simulated a single vehicle and then estimated the distance l and angle γ from the mobile robot to a wireless network access point, we used an exponential path-loss coefficient $\alpha = 2$, and the gaussian variable ν was assumed to have a normal variance of 3 dBm.

The estimated robot trajectory is shown in Figure (5.5) The estimated position error is larger for large distance l, because of the logarithmic relationship between the distance and the estimated RSS.

The next figures, (5.8,5.9,5.10,5.11) show respectively, the trajectories, the RSS, the distance, the relative angle and the distance estimation error for a robot formation of three vehicles $\mathscr{G} = \{\{1,2\}, (0,1), (1,2)\}, l_{01} = l_{12} = 100m, \gamma_{01} = \gamma_{12} = \pi/4$.

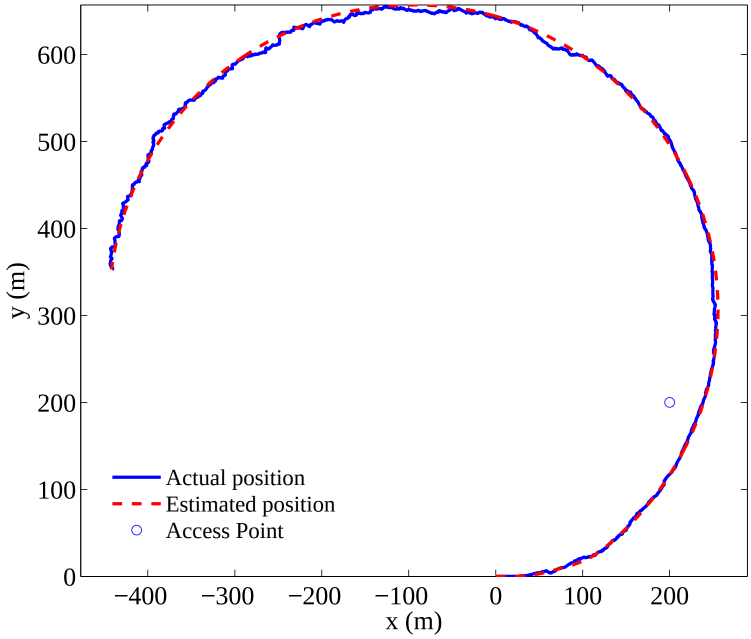


Fig. 5.5: Position estimation.

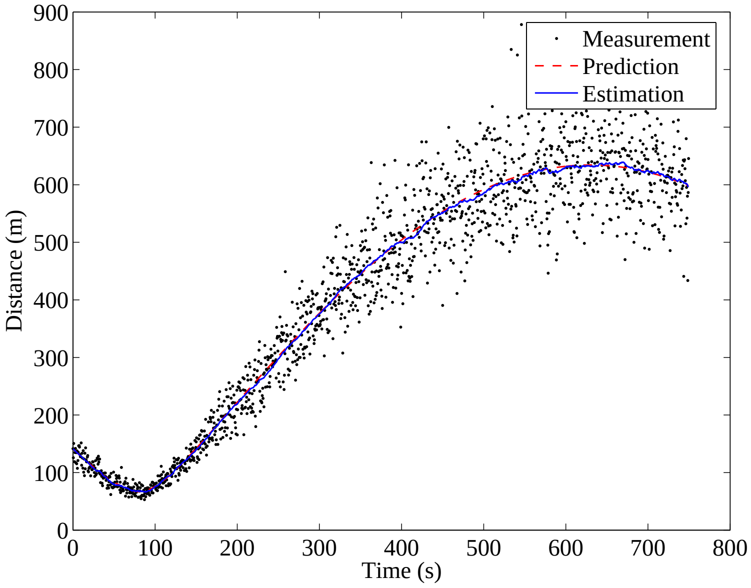


Fig. 5.6: Distance estimation. Circular trajectory example

5.3.2 Experimental Validation

In order to obtain a first validation of the proposed estimation strategy, we performed a simple experiment where the robot is displaced with a constant angle $\gamma = 0$ with respect to the access point (Figs. 5.3 and 5.12).

First, we determined experimentally the α and R_0 values of equation (5.2). Then, the real-time estimator for the distance between the robot and the access point was implemented. Figures (5.13) and (5.14) show, respectively, the estimations for the RSS and distance.

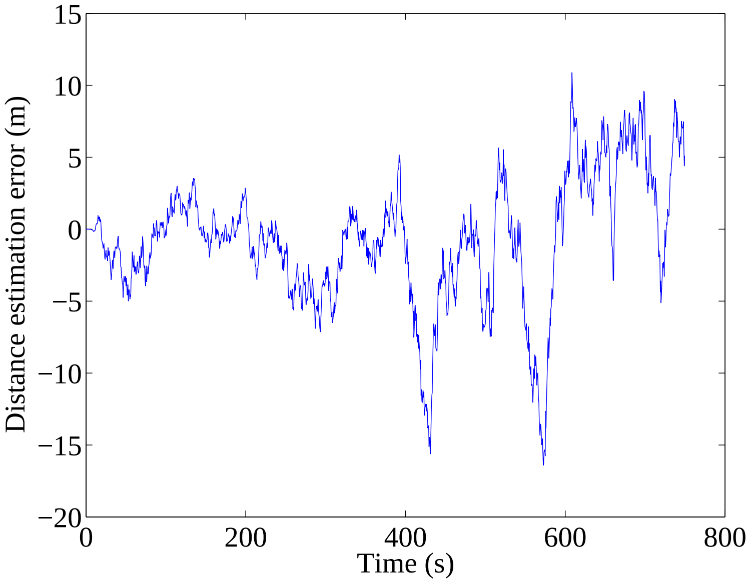


Fig. 5.7: Distance error. Circular trajectory example

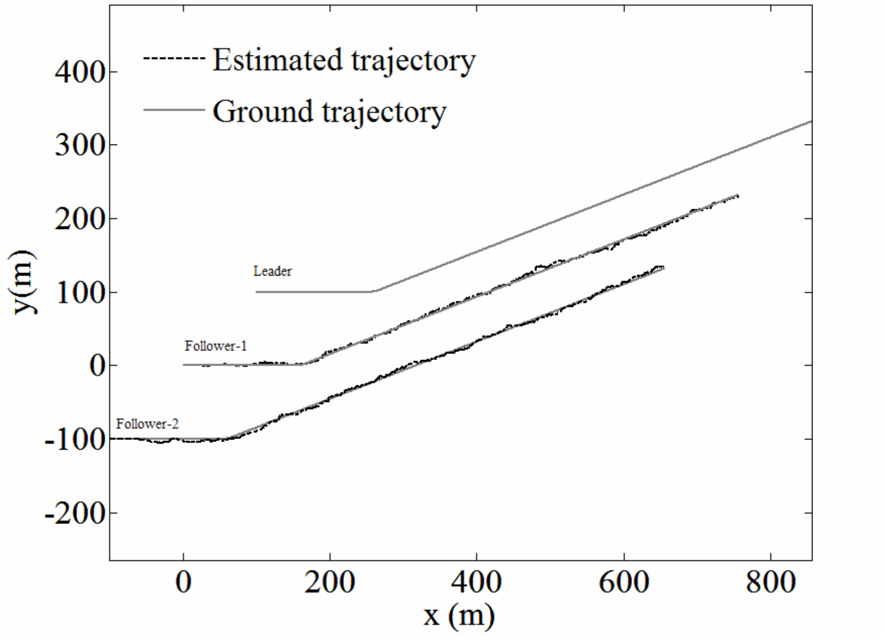


Fig. 5.8: Position trajectory estimation. $\mathscr{G} = \{\{1,2\},(0,1),(1,2)\}, l_{01} = l_{12} = 100m, \gamma_{01} = \gamma_{12} = \pi/4$. We suppose that the RSS is measured between the nodes (0,1) and (1,2)

Therefore, the RSS and distance estimation Figures (5.13,5.14) show that the random RSS signal is reduced by the EKF. A difference between the prediction and estimation is observed, which could be produced by a dynamic model mismatch.

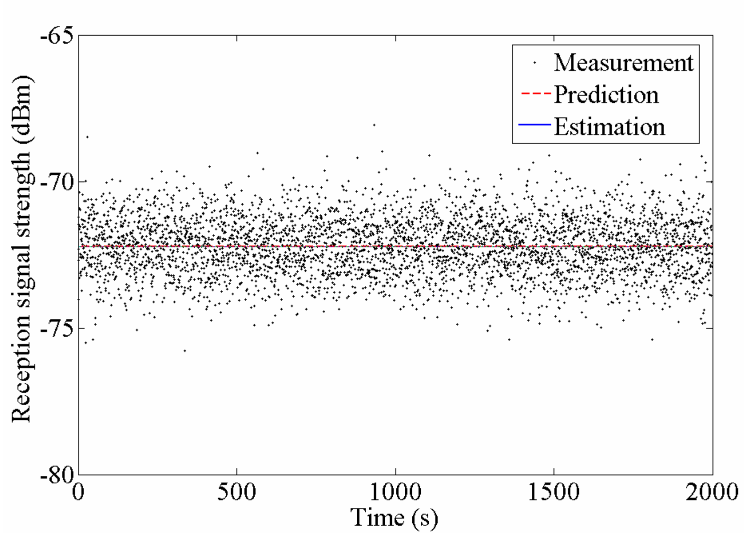


Fig. 5.9: Leader-Follower-1 RSS estimation

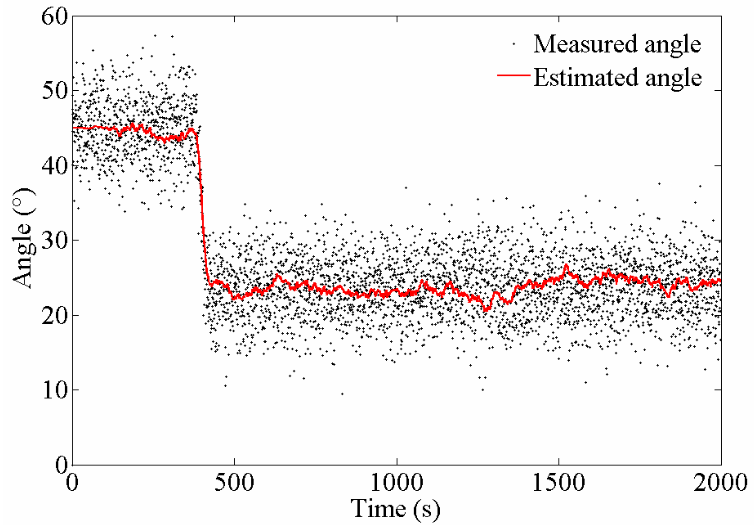


Fig. 5.10: Leader-Follower-1 relative angle γ

5.4 Securing Link Communication

The quality of the received WiFi signal is a function of the power, distance and transmission medium. Every robot can measure the RSS by using its WiFi devices. In order to avoid communication losses, we have to maintain the received signal power above a security level that ensures the quality of the communication for each robot. A constraint of minimal signal power reception must be included in the solution strategy.

To maintain the communication links between robots, we propose to include a term in the performance index that penalizes the loss of power in the received signal. We define the following barrier function:

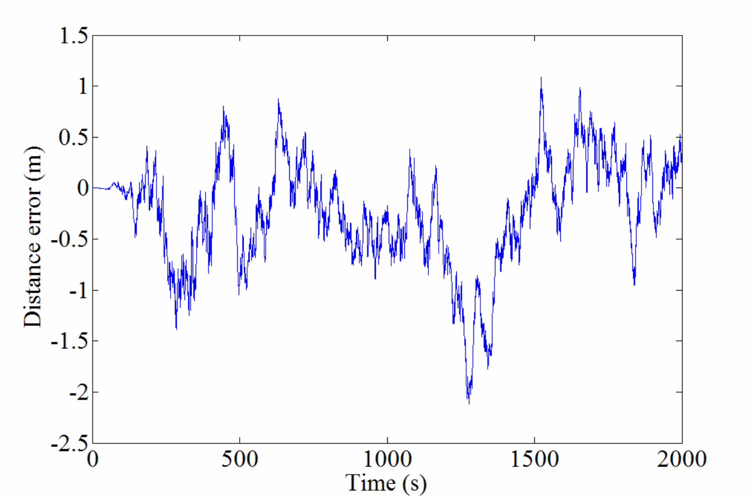


Fig. 5.11: Leader-Follower-1 distance estimation error

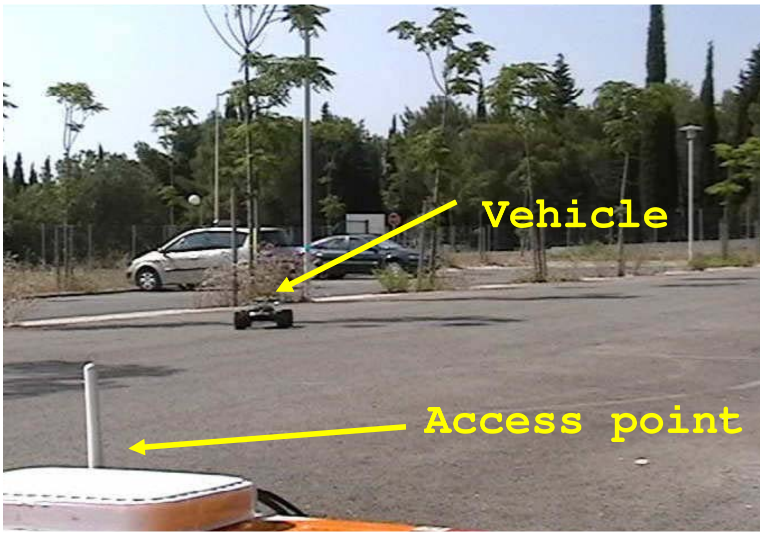


Fig. 5.12: Experimental setup

$$J_j^{com} = \frac{1}{RSS - RSS_{min}}, \tag{5.20}$$

where RSS can be reconstructed by using the propagation model (5.2), and the relative distance $l_{ij}(t)$. We can approximate the value of the RSS over a predicted trajectory and penalize it when it gets close to the predetermined minimal secure signal level RSS_{min}.

5.5 Final remarks

The WiFi network is proposed as a positioning system. The signal RSS and its direction and angle of arrival are measured and these RSS measurements are fused with the vehicle odometric sensors using EKF. This gives an estimation of

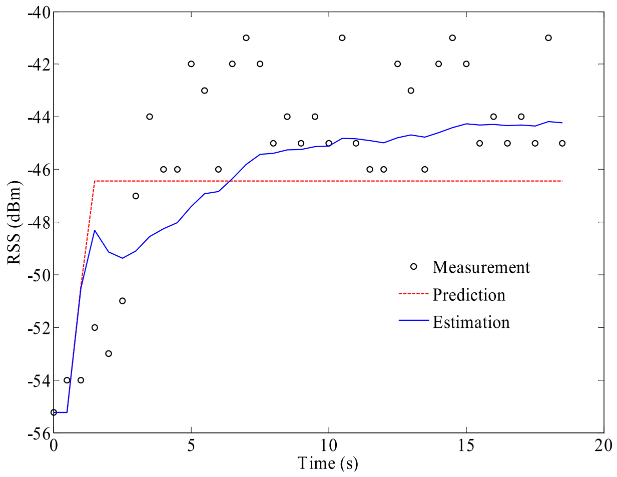


Fig. 5.13: RSS estimation

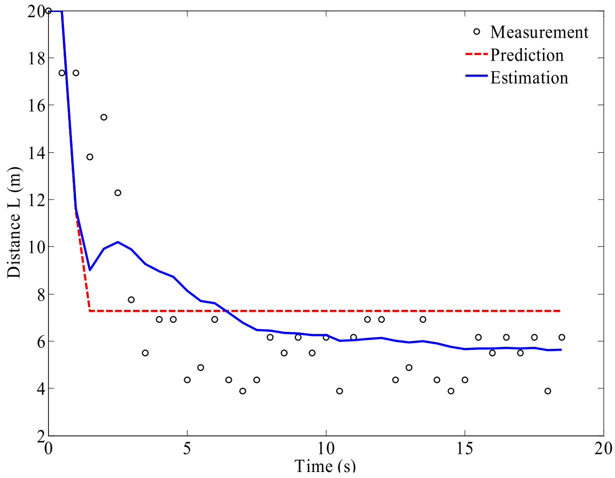


Fig. 5.14: Distance estimation

the relative positions between the robots. The proposed strategy was validated by numerical simulations and experimental measurements. Research is ongoing to study the practical implementation of this device.

6

Single Vehicle Closed-Loop Control

6.1 Introduction

Although the reference trajectories are feasible trajectories for each robot model, a feedback controller to handle model uncertainty is needed. For example, tire slipping and wheel deformability or flexibility are not considered in the pure rolling constraints of the nonholonomic car-like model. Also, friction and nonlinearities in the motors are not modelled. For large slipping conditions, Ackermann [99] proposed a robust control strategy to deal with car skidding. In his work, Ackermann described car control as two separate tasks: path following and disturbance attenuation. In the first task, the driver keeps the car in the planned trajectory; in the second, a controller is proposed to compensate the disturbances. In the present work, as in the Ackermann model, we propose a two-degree-of-freedom control for each vehicle. The first task of this control is to generate a feasible trajectory using the decentralized trajectory planning algorithm described in earlier chapters. Then, we include a second task to compensate disturbances and unmodelled dynamics.

In the trajectory tracking problem, the robot must follow the desired Cartesian path with a specified timing law. The path tracking problem thus consists of stabilizing to zero the two-dimensional Cartesian error **e** (see Fig. 6.1) by using both control inputs.

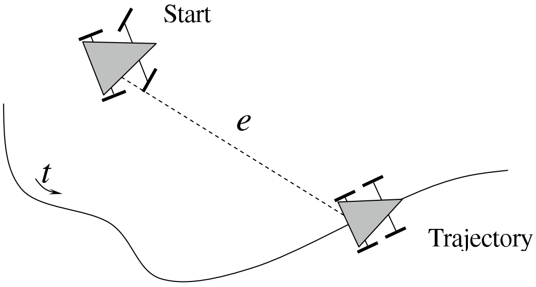


Fig. 6.1: Trajectory tracking problem

In Figure 6.2 , we can see the general two-degree-of-freedom control scheme. The relative position is between the leader i and the follower j; $\mathbf{r}_{ij}$ is compared with the planned one $\mathbf{r}^d_{ij}$, and the planned control $\mathbf{u}^d_j$ is compensated by $\delta_\mathbf{u}$ if any state error exists $\varepsilon_\mathbf{r} = \mathbf{r}^d_{ij} - \mathbf{r}_{ij}$.

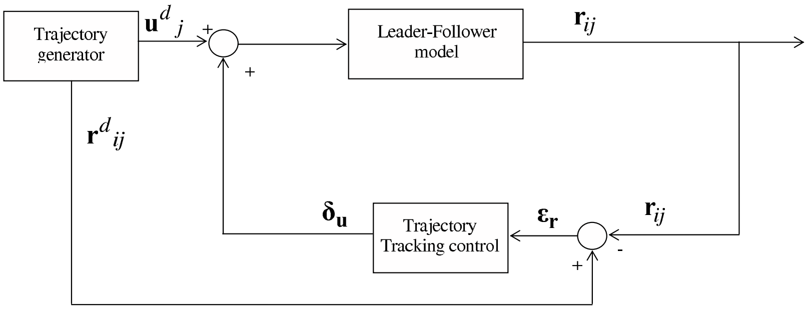


Fig. 6.2: Trajectory generation and tracking.

Many control strategies have been proposed to solve the problem of path tracking in nonholonomic vehicles, including Lyapunov's direct method [100], approximate linearization [101], and static input-output feedback linearization [102]. For robust control, sliding-mode strategies have been applied to the trajectory tracking problem. Benalia *et al* [103], used the advantage of the differential flatness and high-order sliding-mode control to deal with trajectory planning and robust tracking in a car-like robot. Chwa [104] proposed a sliding-mode control method for wheeled-mobile robots in polar coordinates. Also, Defoort *et al* [33], applied an integral sliding mode to the control of unicycle vehicles. Because the linearization of a nonholonomic system about non-stationary trajectories is a controllable system [30], other classical control strategies have been applied to control, like proportional, integral control, PI, or linear quadratic regulators, LQR.

In this chapter, we propose a closed-loop control strategy for the trajectory tracking problem, based on the linear optimal quadratic control strategy LQR. We use the linearized robot model around the reference trajectory and we then define a quadratic performance index for the system. Last, the error of the relative distance between the leader and its follower is asymptotically stabilized.

6.2 Trajectory Tracking Control

The trajectory tracking problem can be defined as follow: Given a feasible trajectory, of the relative position between the leader i and its follower j: $\mathbf{r}^d_{ij}(t)$ with $t \in \mathbb{R}^+$, regulate the state error $\varepsilon_\mathbf{r} = \mathbf{r}_{ij}(t) - \mathbf{r}^d_{ij}(t)$ asymptotically to the origin, subject to the leader-follower model (3.7): $\dot{\mathbf{r}}_{ij} = \mathbf{f}_{ij}(\mathbf{r}_{ij}, \mathbf{v}_j)$.

To solve this problem, we propose to linearize the model system (3.7) around the feasible trajectory $\mathbf{r}^d_{ij}(t)$, then regulate the error $\varepsilon_\mathbf{r}$ to the origin by using a optimal linear quadratic feedback regulation (LQR).

In 1960 Kalman, [105],introduced an integral performance index that had a quadratic penalty on output errors and control magnitudes, and he used the calculus of variations to show that the optimal controls were linear feedbacks of the state variables.

The general LQR problem can be defined as follow: Given the quadratic performance index:

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \tag{6.1}$$

and the linear system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{6.2}$$

stabilize $\mathbf{x}$ to the origin minimizing the performance index J.

Kalman showed that the solution to this problem can be obtained by means of a linear feedback of state variables:

$$\mathbf{u} = -\mathbf{K}\mathbf{x}, \tag{6.3}$$

where the matrix gain $\mathbf{K}$ is obtained by solving the Ricatti equation:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{B}\mathbf{P}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0, \tag{6.4}$$

where $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$

The problem has the following limitations: the pair $(\mathbf{A}, \mathbf{B})$ has to be controllable, the weighs matrix $\mathbf{Q} > 0, \mathbf{R} \geq 0$.

6.2.1 Linearized Leader-Follower Model

Given a feasible trajectory $\{\mathbf{r}_i j^d(t), \mathbf{u}_j^d\}$, with $t \in \mathbb{R}^+$, we can linearize the leader-follower model (3.7), around this trajectory as:

$$\dot{\tilde{\mathbf{r}}}_{ij} = \mathbf{A}(t)\, \tilde{\mathbf{r}}_{ij} + \mathbf{B}\tilde{\mathbf{u}}_j, \tag{6.5}$$

where: $\tilde{\mathbf{r}}_{ij} = \mathbf{r}_{ij} - \mathbf{r}_{ij}^d$, and $\tilde{\mathbf{u}}_j = \mathbf{u}_j - \mathbf{u}_j^d$, and the matrix $\mathbf{A}(t)$ is:

$$\mathbf{A} = \begin{pmatrix} \frac{\partial \dot{\gamma}_{ij}}{\partial \gamma_{ij}} & \frac{\partial \dot{\gamma}_{ij}}{\partial l_{ij}} & \frac{\partial \dot{\gamma}_{ij}}{\partial \theta_j} & \frac{\partial \dot{\gamma}_{ij}}{\partial \xi_j} & \frac{\partial \dot{\gamma}_{ij}}{\partial v_j} & \frac{\partial \dot{\gamma}_{ij}}{\partial \omega_j} \\ \frac{\partial \dot{l}_{ij}}{\partial \gamma_{ij}} & \frac{\partial \dot{l}_{ij}}{\partial l_{ij}} & \frac{\partial \dot{l}_{ij}}{\partial \theta_j} & \frac{\partial \dot{l}_{ij}}{\partial \xi_j} & \frac{\partial \dot{l}_{ij}}{\partial v_j} & \frac{\partial \dot{l}_{ij}}{\partial \omega_j} \\ \frac{\partial \dot{\theta}_j}{\partial \gamma_{ij}} & \frac{\partial \dot{\theta}_j}{\partial l_{ij}} & \frac{\partial \dot{\theta}_j}{\partial \theta_j} & \frac{\partial \dot{\theta}_j}{\partial \xi_j} & \frac{\partial \dot{\theta}_j}{\partial v_j} & \frac{\partial \dot{\theta}_j}{\partial \omega_j} \\ \frac{\partial \dot{\xi}_j}{\partial \gamma_{ij}} & \frac{\partial \dot{\xi}_j}{\partial l_{ij}} & \frac{\partial \dot{\xi}_j}{\partial \theta_j} & \frac{\partial \dot{\xi}_j}{\partial \xi_j} & \frac{\partial \dot{\xi}_j}{\partial v_j} & \frac{\partial \dot{\xi}_j}{\partial \omega_j} \\ \frac{\partial \dot{v}_j}{\partial \gamma_{ij}} & \frac{\partial \dot{v}_j}{\partial l_{ij}} & \frac{\partial \dot{v}_j}{\partial \theta_j} & \frac{\partial \dot{v}_j}{\partial \xi_j} & \frac{\partial \dot{v}_j}{\partial v_j} & \frac{\partial \dot{v}_j}{\partial \omega_j} \\ \frac{\partial \dot{\omega}_j}{\partial \gamma_{ij}} & \frac{\partial \dot{\omega}_j}{\partial l_{ij}} & \frac{\partial \dot{\omega}_j}{\partial \theta_j} & \frac{\partial \dot{\omega}_j}{\partial \xi_j} & \frac{\partial \dot{\omega}_j}{\partial v_j} & \frac{\partial \dot{\omega}_j}{\partial \omega_j} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{\partial \dot{\gamma}_{ij}}{\partial u_{1j}} & \frac{\partial \dot{\gamma}_{ij}}{\partial u_{2j}} \\ \frac{\partial \dot{l}_{ij}}{\partial u_{1j}} & \frac{\partial \dot{l}_{ij}}{\partial u_{2j}} \\ \frac{\partial \dot{\theta}_j}{\partial u_{1j}} & \frac{\partial \dot{\theta}_j}{\partial u_{2j}} \\ \frac{\partial \dot{\xi}_j}{\partial u_{1j}} & \frac{\partial \dot{\xi}_j}{\partial u_{2j}} \\ \frac{\partial \dot{v}_j}{\partial u_{1j}} & \frac{\partial \dot{v}_j}{\partial u_{2j}} \\ \frac{\partial \dot{\omega}_j}{\partial u_{1j}} & \frac{\partial \dot{\omega}_j}{\partial u_{2j}} \end{pmatrix}, \tag{6.6}$$

Then, applying the partial derivatives to the system 3.7, we obtain the linearized system and control matrices, $\mathbf{A}, \mathbf{B}$, which are expressed as follows:

$$\mathbf{A} = \begin{pmatrix} 0 & v_j^d \sin \gamma_{ij}^d & v_j^d \sin \gamma_{ij}^d & 0 & -\cos \gamma_{ij}^d & 0 \\ -\frac{v_j^d}{{l_{ij}^d}^2} \left(\sin {\gamma_{ij}^d}^2 \right) & \frac{v_j^d}{l_{ij}^d} \left(\cos \gamma_{ij}^d \right) & \frac{v_j^d}{l_{ij}^d} \left(\cos \gamma_{ij}^d \right) & \frac{v_j^d}{L} \left(1 + \tan^2 \xi_j^d \right) & \frac{1}{l_{ij}^d} \sin \gamma_{ij}^d + \frac{1}{L} \tan \xi_j^d & 0 \\ 0 & 0 & 0 & \frac{v_j^d}{L} \left(1 + \tan^2 \xi_j^d \right) & \frac{1}{L} \tan \xi_j^d & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_v} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_\omega R} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau_v} & 0 \\ 0 & \frac{1}{\tau_\omega} \end{pmatrix},$$

6.2.2 Controllability Conditions

We can note that the linearized system (6.5) is time-varying through the dependence on time of the trajectory. As a consequence, the controllability analysis is more involved than in the time-variant case.

Definition 6.2.1. Controllability
The time-varying system $\dot{x} = A(t)x + B(t)u$ is controllable if for all $(a,b) \in (\mathbb{R}^n \times \mathbb{R}^m)$, there is a trajectory $(x,u) \in C^0([T_0,T_1];\mathbb{R}^n) \times C_d^0([T_0,T_1];\mathbb{R}^m)$, with:

$$x(T_0) = a, \quad and \quad x(T_1) = b,$$

By defining:

$$B_i \in C^\infty([T_0,T_1];\mathbb{R}^m \times \mathbb{R}^n),$$

par:

$$\begin{array}{rcl} B_0 & = & B \\ B_i & = & AB_{i-1} - \frac{d}{dt}B_{i-1} \end{array}, \tag{6.7}$$

The following theorem can be obtained [106], where $ev\{M\}$ defines the sub-vectorial space of $\mathbb{R}^n$ generated by $M \subset \mathbb{R}^n$.

Theorem 6.2.2. *[106]*
If there is $t \in [T_0,T_1]$*, for which:*

$$ev\{B_i(t)v; v \in \mathbb{R}^m \;; i \in \mathbb{N}\} = \mathbb{R}^n,$$

The system $\dot{x} = A(t)x + B(t)u$ *is then controllable.*

For the linearized system (6.5), we can note that the control matrix B is a constant, then from equation (6.7) the B_i values can be defined as:

$$\{B_0 = B, B_1 = AB, B_2 = A^2B, ...B_i = A^iB\}, i \in \mathbb{N} \tag{6.8}$$

Then, if we apply the theorem (6.2.2) to the system (6.5) with B time-invariant, we can evaluate the controllability for $i = 5$ by evaluating the rank of the following matrix:

$$rank\begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A^2B} & \mathbf{A^3B} & \mathbf{A^4B} & \mathbf{A^5B} \end{bmatrix} = 6, \tag{6.9}$$

We can verify that this matrix has two nonzero 6×6 minor values D_1,D_2:

$$D_1 = \frac{{v_j^d}^2\left(1+\tan^2\left(\xi_j^d\right)\right)}{{\tau_v}^3{\tau_\omega}^3L^2{l_{ij}^d}^2}$$
$$\left(2\,\cos^2\left(\gamma_{ij}^d\right)\tan\left(\xi_j^d\right)l_{ij}^d+\sin^3\left(\gamma_{ij}^d\right)L+2\,\sin^2\left(\gamma_{ij}^d\right)\tan\left(\xi_j^d\right)l_{ij}^d\right), \tag{6.10}$$

$$D_2 = \frac{{v_j^d}^4\left(1+\tan^2\left(\xi_j^d\right)\right)}{tw^3tv^4{l_{ij}^d}^4L^3}\sin^2\left(\gamma_{ij}^d\right)$$
$$\left(2\,\cos^2\left(\gamma_{ij}^d\right)\tan\left(\xi_j^d\right)l_{ij}^dL+\sin^3\left(\gamma_{ij}^d\right)L^2+4\,\sin^2\left(\gamma_{ij}^d\right)L\tan\left(\xi_j^d\right)l_{ij}^d+4\,\sin\left(\gamma_{ij}^d\right)\tan^2\left(\xi_j^d\right){l_{ij}^d}^2\right) \tag{6.11}$$

Therefore, the pair $(\mathbf{A},\mathbf{B})$ will not be controllable when D_1 and D_2 are nulls, $(D_1=0, D_2=0)$. The first value D_1 is zero if:

$$\begin{cases} v_j^d = 0 & or \\ \{\xi_j^d = 0, \gamma_{ij}^d = 0\} & or \\ \{\xi_j^d = 0, \gamma_{ij}^d = \pi\} & or \\ L = -\frac{2\tan\left(\xi_j^d\right)l_{ij}^d}{\sin^3\left(\gamma_{ij}^d\right)} & \end{cases}, \tag{6.12}$$

and the second value D_2 is zero for:

$$\begin{cases} v_{ij}^d = 0 & or \\ \gamma_{ij}^d = 0 & or \\ L = \frac{\left(-\sin^2\left(\gamma_{ij}^d\right)-1+\sqrt{-3\sin^4\left(\gamma_{ij}^d\right)+2\sin^2\left(\gamma_{ij}^d\right)+1}\right)\tan\left(\xi_j^d\right)l_{ij}^d}{\sin^3\left(\gamma_{ij}^d\right)} & \end{cases}. \tag{6.13}$$

Accordingly, the system 6.7 is not controllable in the interception of solutions 6.12 and 6.13; this interception is expressed by the following expression:

$$\begin{cases} v_{ij}^d = 0 & or \\ \{\xi_j^d = 0, \gamma_{ij}^d = 0\} & or \end{cases}, \tag{6.14}$$

and:

$$\begin{aligned} L = \quad & -\frac{2\tan\left(\xi_j^d\right)l_{ij}^d}{\sin^3\left(\gamma_{ij}^d\right)} & = \quad & \frac{\left(-\sin^2\left(\gamma_{ij}^d\right)-1+\sqrt{-3\sin^4\left(\gamma_{ij}^d\right)+2\sin^2\left(\gamma_{ij}^d\right)+1}\right)\tan\left(\xi_j^d\right)l_{ij}^d}{\sin^3\left(\gamma_{ij}^d\right)} \\ \Rightarrow \quad & -2 & = \quad & -\sin^2\left(\gamma_{ij}^d\right)-1+\sqrt{-3\sin^4\left(\gamma_{ij}^d\right)+2\sin^2\left(\gamma_{ij}^d\right)+1} \\ \Rightarrow \quad & \gamma_{ij}^d = \pm\pi/2, & & \end{aligned} \tag{6.15}$$

Consequently, the linearized system 6.7 is controllable as long as $v_j^d \neq 0$, $\{\xi_j^d \neq 0, \gamma_{ij}^d \neq 0\}$ and $\gamma_{ij}^d \neq \pm\pi/2$. These controllability conditions can be included in the path planning algorithm; that is, we will obtain only trajectories that satisfy the linearized system controllability conditions.

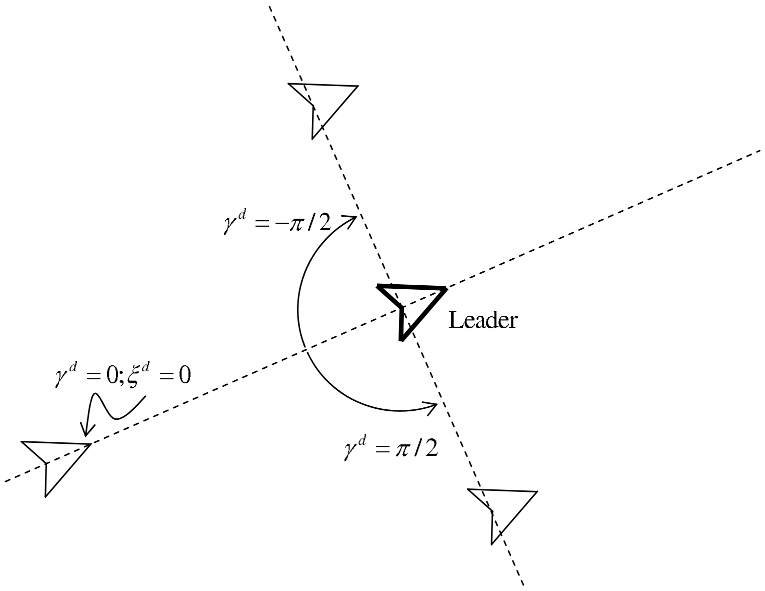


Fig. 6.3: Linearized system, non-controllable condition. $\gamma^d = \pm\pi/2$ and $(\gamma^d = 0, \xi^d = 0)$

6.2.3 Simulations

In the following figures, we can see the LQR implementation results. For a nominal reference trajectory, we applied a gain scheduled LQR; that is at each sampling time we solve the Ricatti equation (6.4) for the linearized system (6.7) to obtain the feedback gain.

In a first simulation we see the performance of the LQR controller to stabilize the vehicle in a desired linear trajectory. The measured variables are the linear and angular velocities (v_j, ω_j), the orientation angle for the leader and the follower: θ_j, θ_i, and the relative distance and angle between the leader i and follower j: ł$_{ij}, \gamma_{ij}$.

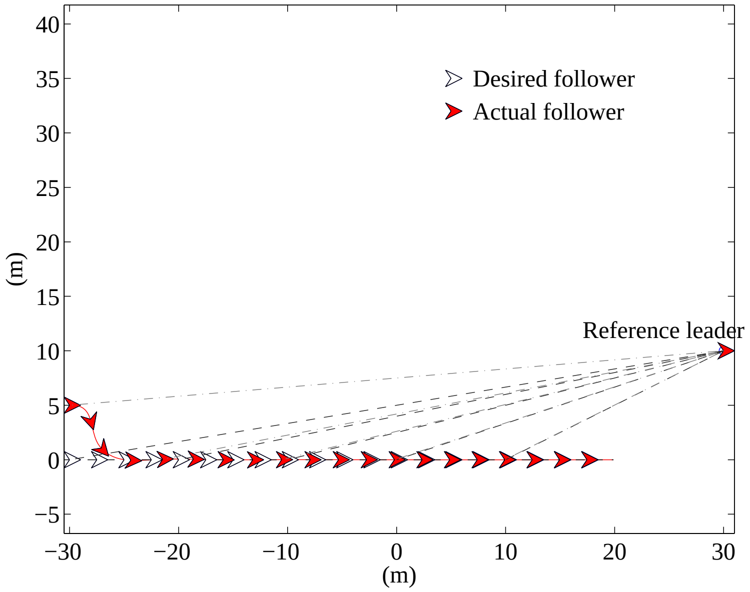


Fig. 6.4: Closed-loop relative distance l_{ij}.

The Figure 6.4, shows the vehicle trajectory stabilizing around a desired feasible straight trajectory; this reference trajectory is defined by the relative distance $l_{ij}(t)$ and angle $\gamma_{ij}(t)$. We can see how the vehicle trajectory asymptotically tracks the reference path from a initial condition $\mathbf{r}_{ij}(0) \neq \mathbf{r}^d_{ij}(0)$. The error variation over the time shows that the vehicle trajectory converge to the desired one in a finite time (see Fig. 6.5).

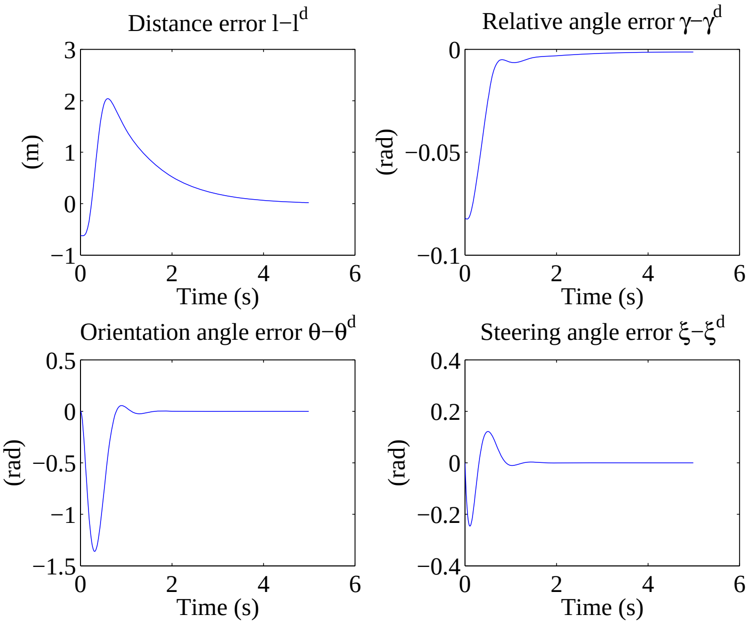


Fig. 6.5: Closed-loop relative angle γ_{ij}.

The LQR controller is tested for a circular reference path (see Fig. 6.6). The controlled variables error figures (6.7) show the stabilization around the desired trajectory by eliminating the initial position error.

6.3 Final Remarks

In this chapter, we implemented an LQR to stabilize a vehicle in the reference path. We obtained the controllability conditions for the linearized leader-follower model.

The control proposed in this section is based in the linearized vehicle model, we linearized about a feasible trajectory, such of that, the stability conditions are only guarantee if the robot current trajectory is near to the feasible trajectory. The proposed trajectory computes a each sampling time a trajectory from the current robot position, this characteristic guarantees that the follower robot will be near to the desired trajectory with a small variation. Other robust strategies, based on the nonlinear model can be applied to the trajectory tracking problem, like integral sliding mode control [33], but by proposing the LQR we validated the general multi-robot control strategy, i.e. the use of a feasible path and then the stabilization around this trajectory. One of the main drawbacks of the proposed closed-loop control is the sensitivity of the robot position with respect to the relative angle γ_{ij}

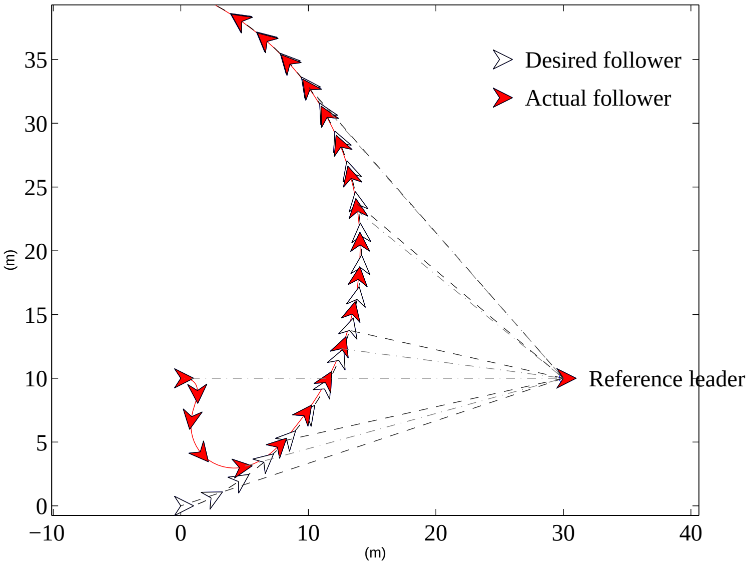


Fig. 6.6: Closed-loop linear velocity v_j.

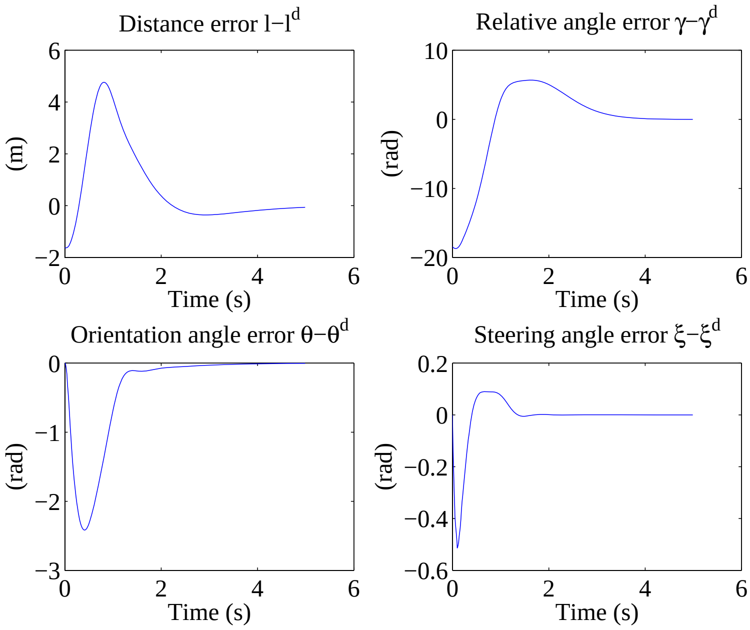


Fig. 6.7: Closed-loop steering angular velocity ω_j.

and distance l_{ij}. For large relative distances, small errors in the desired relative angle produce large variations in the x, y position, and more precise and robust control has to be implemented. Ongoing works are studying more complex control strategies to stabilize the leader-follower model in the feasible reference path.

7

Conclusions

In this work, a novel decentralized method for generation of an adaptive trajectory was proposed to control the formation of nonholonomic robots with state and input constraints. We proposed a path generation method based on the optimization of a general performance index. This strategy can be applied to different kind of robotic formation, like unmanned aerial, terrestrial or underwater vehicles. Furthermore, by defining a general performance index we can optimize different tasks, like energy or time minimization, or other specific path constraints, for example the inclusion of currents conditions in the path planning for underwater multi-robot systems.

This work deals with terrestrial vehicles, in this kind of vehicles the obstacle avoidance is one main task. Such of that, we include into the performance index definition a *obstacle avoidance* term based on DVZ concept, also we include a term for the time minimization an other term to ensure the communication links between robots. The control strategy permits to control the robot formation in unknown environments, by allowing each robot to autonomously avoid the obstacles based only on the local information of its sensors. Due to the decentralized nature of the control configuration, this methodology can be applied for a large number of robots without any additional computational effort. The formation obtained is not rigid, but is flexible to avoid obstacles. The desired robot formation is achieved through the decentralized leader-follower approach, in which every robot is positioned by using only its local sensor information. Finally, sensing the level of signal reception of the Wifi network allows the leader-follower positioning strategy to be applied, even if GPS robot data are not available or if the quality of the communication link decreases. The validation of the proposed strategy, shows that it is a secure general solution to real application of control of nonholonomic constrained ground vehicles robots formation in unknown environments.

7.1 Multi-robot Coordination

In the area of multi-robot coordination, this thesis proposes a distributed leader-follower relationship. One of main advantages of this configuration is that it takes into account the nonholonomic constraints and actuator (motor) limitations of car-like vehicles. Most of the previous works in this area were based on simplified models like the unicycle and did not take into account actuator saturations. The use of a feasible sequentially programming algorithm provides the trajectory generator an optimal path , or at least a feasible usable trajectory. We present a particular performance index for trajectory optimization, and this index can be generalized to include many criteria based on the mission demands, like energy minimization or other desired path constraints. We present an analysis of the trajectory generation convergence and prove that for leader trajectory all the followers can track in a finite time its desired position.

7.2 Obstacle Avoidance

To deal with the obstacles avoidance, we include the main concept of the DVZ method into the trajectory generation. This method has been widely tested and applied in nonholonomic mobile systems. The proposed obstacle avoidance strategy is based on four ultrasound sensors, and we present simulations and validations showing that with very simple and limited information we can deal with complex environments.

As we used a general optimization algorithm, we can include it in other strategies to avoid obstacles. In the future, comparative studies of obstacle avoidance approaches should be carried out.

7.3 Communication as Positioning System

We validate the use of a wireless communication network to secure the links between robots. A model to estimate the relative positions between the formation robots is proposed and validated by simulations and experiments. The positions are estimated using the received signal strength, RSS, and the direction of the maximal RSS. This last task was not studied experimentally, but some simulations are presented to validate the general approach.

Wireless RSS can be used not only to ensure the links between robots, but also to build an obstacle avoidance strategy. If a robot is following the maximal RSS and an obstacle modifies the RSS field, the robot can use this information to avoid the obstacle.

7.4 Outgoing Work

This research is focussed on unmanned terrestrial vehicles. Ongoing works are studying the performance of the proposed control architecture when it is applied to other types of unmanned vehicle formations, like homogeneous or heterogeneous sets of terrestrial, aerial and underwater mobile robots. With these approaches, more complex optimization performance indexes, or cost functions could be included, such as the minimization (maximization) of cooperation between robots for a specific task or mission.

The proposed strategy is based on the optimal trajectory for each single robot; thus the optimal conditions for the whole robot set can be studied. For example, how can we obtain a global condition for the entire formation from a local optimal condition? The main problem posed by this study is how to simultaneously impose a decentralized control strategy and a global constraint.

In the context of position measurement, future research can be carried out in the area of fusing GPS, WiFi and odometric sensor signals. This fusion would compensate the errors of each system and could improve security. Also, when a WiFi network is used as a positioning system, the effects of obstacles on the wireless signal propagation could be studied, and the data might be applicable to the problem of obstacle avoidance.

For the problem of path tracking, other techniques could be applied, such as more robust control strategies like higher order sliding-mode control.

8

Résumé

Ces travaux concernent l'élaboration d'une stratégie de commande décentralisée réactive pour une flottille de robots mobiles terrestres. Cette stratégie de commande est basée sur un contrôle décentralisé qui s'appuie sur le principe Leader-Follower utilisant à la fois des informations de positionnement absolu (GPS) et relatif entre véhicules (niveau de réception des liens WiFi) ainsi que des informations d'existence d'obstacles de proximités (capteurs ultra-sons). Cette méthode permet d'intégrer et d'optimiser à chaque instant ces diverses contraintes afin de générer un chemin faisable. Mais également de maintenir la flotille dans une forme géometrique donnée, avec un niveau de réception des transmissions entre les véhicules minimal, tout en évitant d'éventuels obstacles.

Mots clefs

Commande collaborative, systèmes plats, génération de trajectoire, commande décentralisée

8.1 Introduction

Les systèmes multi-robots constituent un champ important de la recherche en robotique. En effet ils ouvrent de larges perspectives dans les domaines de l'exploration ou des missions de recherche et de secours des personnes, qui necessitent une couverture de l'espace importante. Pour ce faire, le partage d'informations capteurs comme la localisation de l'ensemble des véhicules ou les retours vidéo et sonore vers un opérateur distant assurent une couverture rapide et optimale de la recherche. D'autres types de missions collaboratives ont déjà été envisagées tels que le sauvetage par des systèmes mutli-robots [11], [12], l'exploration coopérative [13], [14], [15], ou les jeux d'équipe [16],[17]. Pour réaliser ces tâches, différentes techniques de contrôle coopératif ont déjà été abordées, par exemple des stratégies centralisées ou décentralisées. Des études sur la classification des systèmes multi-robots ont été présentées par Farinelli *et al* [1] et par Cao *et al* [18]. Dans la plupart des cas, les techniques décentralisées s'appuient sur l'utilisation de règles de comportement [107] observées dans la nature, comme celles des colonies de fourmis ou des troupeaux d'oiseaux. Notre travail de recherche s'est focalisé sur l'étude et le développement d'une stratégie de commande. Cette stratégie devra être capable de maintenir une flottille de véhicules non holonomes en formation, avec une contrainte de maintien des liens de transmission inter-véhicules, et cela quelle que soit la structure de l'environnement (bâtiments, murs, obstacles). En effet, la force et l'efficacité d'une flottille de robots est dans le partage des informations, ainsi que dans un déplacement coordonné permanent et réactif. Ces contraintes étant respectées il devient envisageable pour un opérateur distant de téléopérer la flottille (cf. figure (8.1)).

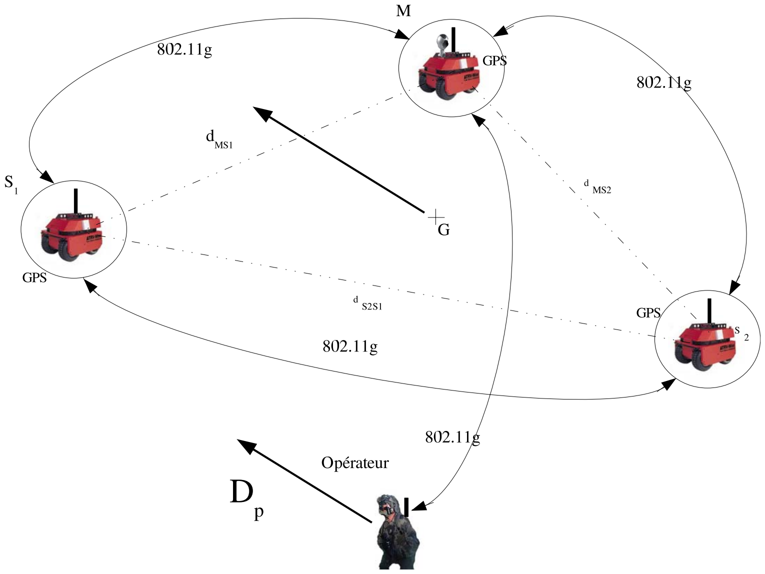


Fig. 8.1: Flottille de robots mobiles

La stratégie de commande que nous avons développée est basée sur le principe d'une commande décentralisée leader-follower proposée dans [108], [44]. Cependant le déploiement d'une flottille de robots d'extérieur dans des environnements urbains par exemple, peut engendrer des difficultés d'une part de localisation avec le GPS mais aussi de coupure des liens de communications entre véhicules. Aussi avons-nous choisi d'utiliser le dispositif de transmission sans fil WiFi en tant que capteur, permettant de mesurer le niveau de réception du lien de transmission d'une véhicule avec son voisin immédiat. Cette nouvelle variable de commande associée à l'évitement local de collision, nous contraint a l'élaboration d'un schéma de commande capable de gérer le positionnement absolu et relatif des véhicules, le niveau de réception des transmissions ainsi que l'évitement de collision.

8.2 Stratégie de commande décentralisée

8.2.1 Introduction

La stratégie de commande décentralisée est basée sur le modèle Leader-Follower (cf. figure (8.2)). Au début de la mission un véhicule est choisi arbitrairement comme leader de la flottille (0). Les autres véhicules sont les followers. La position désirée du leader est définie arbirairement. Les positions désirées des followers sont

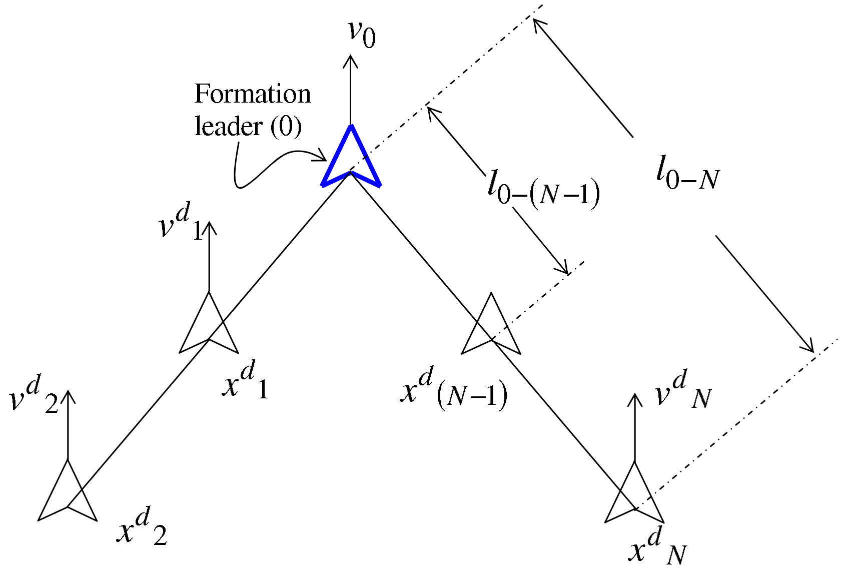


Fig. 8.2: Stratégie de commande décentralisée

définies par la position courante du leader (distance et angle entre le suiveur j et leader i).

Si cette configuration est respectée, alors les véhicules vont suivre le déplacement du leader de la formation (robot 0) (cf. figure (8.2)) à condition que tous les véhicules accèdent à chaque instant à la position de leur leader. La configuration de la formation est définie par les graphes $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, où le sommets $\mathscr{V} = \{1, ...n\}$ représentent les n followers et $\mathscr{E} = \{(i, j) : i, j \in \mathscr{V} \cup \{0\}\}$ les arrêtes définissant les relations leader-followers de la formation.

Les graphes:

$$\mathscr{G}_a = (\{1,2\}, \{(0,1),(0,2)\}); \quad \mathscr{G}_b = (\{1,2\}, \{(0,1),(1,2)\}) \ ,$$

définissent deux formations de robots différentes (cf. figure 8.3).

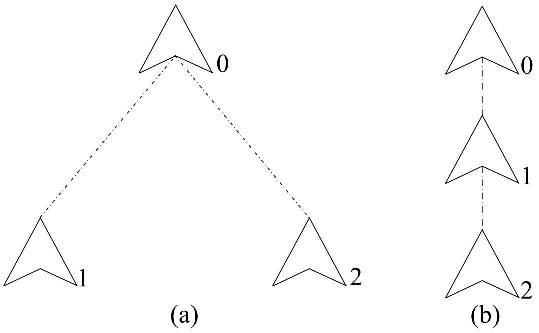


Fig. 8.3: (a) Delta, (b) Linéal

8.2.2 Positionnement leader-follower

Les positions désirées de chacun des robots constituant la flottille, se déduisent des informations de positionnement absolu (GPS) courantes de chacun des véhicules

x_i,y_i, grâce à la connaissance de l'angle relatif γ_{ij} entre le leader i et follower j. L'angle relatif étant mesuré par une antenne sectorielle motorisée pointant le niveau de réception maximal du véhicule leader (cf. figure (8.4)).

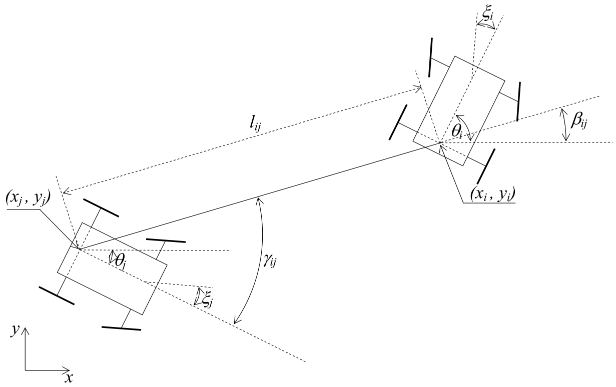


Fig. 8.4: Positionnement Leader-Follower

Chaque véhicule est modélisé par, [30]:

$$\begin{pmatrix} \dot{x}_i \\ y_i \\ \theta_i \\ \xi_i \end{pmatrix} = v_i \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \\ \tan\xi_i/L \\ 0 \end{pmatrix} + \omega_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{8.1}$$

En utilisant l'ensemble de ces informations, on peut écrire les équations des positionnement relatifs entre le leader et les follower:

$$\begin{aligned} \dot{\mathbf{r}}_{ij} = {} & \begin{pmatrix} -\cos(\gamma_{ij}) & 0 \\ \sin(\gamma_{ij})/l_{ij} + \tan(\xi_j)/L & 0 \\ \tan(\xi_j)/L & 0 \\ 0 & 1 \\ -1/\tau_v & 0 \\ 0 & -1/(\tau_\omega R) \end{pmatrix} \mathbf{v}_j + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/\tau_v & 0 \\ 0 & 1/\tau_\omega \end{pmatrix} \mathbf{u}_j + \\ & \ldots + v_i \begin{pmatrix} \cos(\theta_i - \beta_{ij}) \\ \sin(\theta_i - \beta_{ij})/l_{ij} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{8.2}$$

avec, $\mathbf{r}_{ij} = (\ l_{ij} \quad \gamma_{ij} \quad \theta_j \quad \xi_j \quad v_j \quad \omega_j\ )^T$ et $\beta_{ij} = \theta_j - \gamma_{ij}$.

8.2.3 Génération de trajectoires en temps réel

La commande de la flottille de robots mobiles doit être compatible avec l'ensemble des contraintes énoncées ci-dessus, et inclure de plus, les contraintes de non-holonomie de chacun des véhicules et les saturations des actionneurs de chaque véhicule. En effet, si lors d'un déplacement de la formation l'actionneur d'un véhicule atteint la saturation en vitesse ou en direction, un comportement en boucle ouverte apparaîtra risquant de détruire la formation. Aussi est-il indispensable de prendre en compte cette contrainte.

Le générateur de trajectoires proposé utilise le calcul d'une trajectoire optimale prenant en compte les contraintes de saturation ainsi que le modèle du véhicule. Cette trajectoire est calculée pour rejoindre la position désirée à partir de la position courante (cf. figure 8.5)

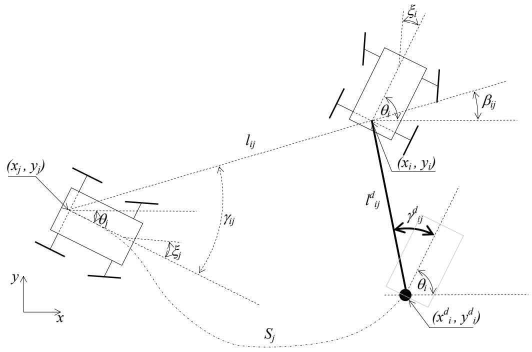


Fig. 8.5: Définition de la trajectoire optimale

La commande optimale est définie à l'instant t_k par :

$$\min_{\mathbf{r_{ij}},\mathbf{u_j}} J_j = \int_{t_0}^{t_f} L\left(\mathbf{r}_{ij}, \mathbf{u}_j\right) dt, \tag{8.3}$$

pour les conditions initiales

$$\mathbf{r}_{ij}(t_0) = \mathbf{r}_{ij}(t_k), \tag{8.4}$$

les conditions finales

$$\mathbf{r}_{ij}(t_f) = \mathbf{r}^d_{ij}(t_k), \tag{8.5}$$

la saturation des commandes

$$(u_{1min}, u_{2min}) \leq (u_{1j}, u_{2j}) \leq (u_{1max}, u_{2max}), \tag{8.6}$$

et les contraintes d'inégalités sur l'état

$$\xi_{min} \leq \xi_j \leq \xi_{max}. \tag{8.7}$$

Pour le modèle du véhicule (éq. 8.1), les variables (x, y) sont connues comme étant les sorties plates du système [72]. En utilisant les sorties plates et ses dérivées, toutes les variables du système peuvent être obtenues par des fonctions algébriques. Cette représentation du modèle dynamique du robot en fonction de ses sorties plates, permettra de réaliser une paramétrisation d'une trajectoire basée sur des B-Splines. Par conséquent, le problème de commande optimale est transformé dans un problème d'optimisation de paramètres. La solution est obtenue en utilisant l'algorithme de programmation non linéaire CFSQP [78]. L'utilisation de CFSQP permet d'obtenir une solution faisable a chaque iteration. Ainsi, si dans un délai de temps déterminé la solution optimale ne peut être calculée, une solution faisable de référence sera néamoins disponible.

En raison du temps de calcul necessaire pour un chemin optimal, on propose un algorithme de gestion de la génération de trajectoire dans le temps. Afin de pouvoir calculer et appliquer les solutions optimales en temps réel. Cet algorithme est inspiré par les travaux de Milam [41], et Van Nieuwstadt & Murray [40]. L'algorithme de génération de trajectoires en temps réel se résume par les actions suivantes:

- On définit le temps t_k par $t_k = t_{k-1} + \tau$, avec $k = 1,2...$, où τ est le temps pour le calcul d'une nouvelle trajectoire.
- Pour l'intervalle $[t_{k-1}, t_k]$, on applique la commande $U_{{k-1}_j}$, et on calcule la solution optimale suivante $U_{k_j} = \left\{\mathbf{u}(t)_j \in \mathbb{R}^2 : t \in [t_k, t_{f_k})\right\}$, $S_{k_j} = \left\{\mathbf{r}_{ij}(t) \in \mathbb{R}^4 : t \in [t_k, t_{f_j}]\right\}$, avec $\mathbf{r}_{ij}(t_{f_k}) = \mathbf{r}^d_{ij}(t_{k-1})$, $\mathbf{r}_{ij}(t_k) \in S_{k-1}$, et $t_{f_j} \geq \tau$, où t_{f_k}, est le temps final de la trajectoire calculée S_{k_j}.

La figure 8.6 ilustre l'application de cet algorithme sur un véhicule.

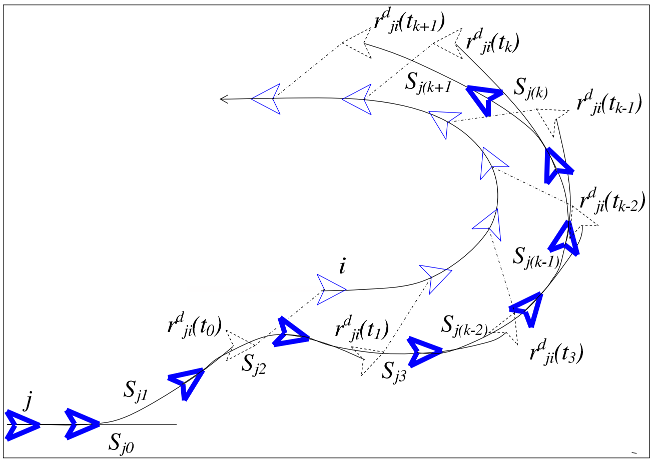


Fig. 8.6: Algorithme de génération de trajectoires en temps réel.

8.2.3.1 Trajectoire optimale à temps minimal

Dans la méthode d'optimisation présentée, différentes fonctions de coût J peuvent être utilisées. Par exemple la minimisation du temps définie par la fonction de coût suivante:

$$J = \int_0^{t_f} dt = t_f. \tag{8.8}$$

L'application de l'algorithme proposé permet d'obtenir une solution au problème de coordination de mouvement d'une flottille de véhicules non holonomes. La figure 8.7 montre la simulation de la commande d'une flottille de trois véhicules. Sur la figure de gauche, le temps du calcul inter-trajectoires a été établi à $\tau = 300ms$, sur la figure de droite, ce temps est de $\tau = 500ms$.

Pour les deux valeurs de temps τ, on obtient une solution faisable, mais la solution pour $\tau = 300ms$ présente un temps de stabilisation plus large qu'avec $\tau =$

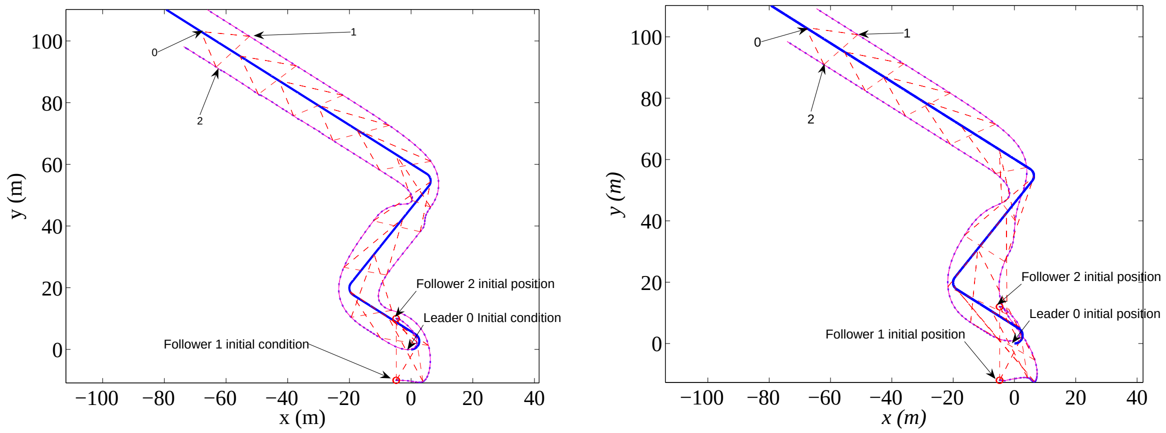


Fig. 8.7: Flotille: $\mathscr{V} = \{1,2\}$, $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Gauche $\tau = 300ms$. Droite $\tau = 500ms$

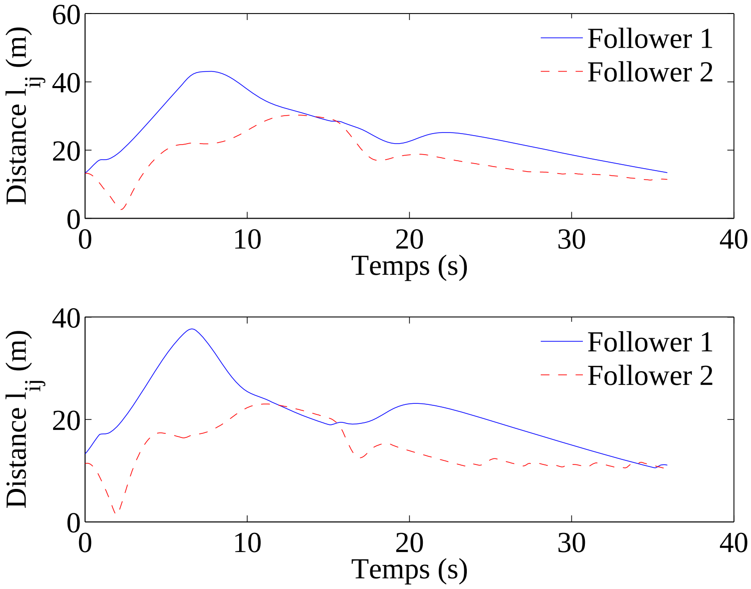


Fig. 8.8: Distance relatif l_{ij}: $\mathscr{V} = \{1,2\}$. $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. En haut $\tau = 300ms$. En bas $\tau = 500ms$

$500ms$. Cette différence dans le temps de stabilisation peut être observée dans les figures 8.8,8.9 qui montrent la stabilisation des variables d'état du système.

Finalement, on peut vérifier que les solutions obtenues respectent les contraintes des saturations, $v_{max} = 6m/s, \xi_{max} = \pi/6$, (cf. figures 8.10, 8.11).

La stratégie de commande proposée étant une méthode décentralisée, on peut l'appliquer pour la commande d'une flottille de n véhicules. La solution pour la commande d'une formation de douze robots mobiles est illustrée par la figure 8.12. Dans cette simulatioin la formation désirée change à l'instant $t = 11s$.

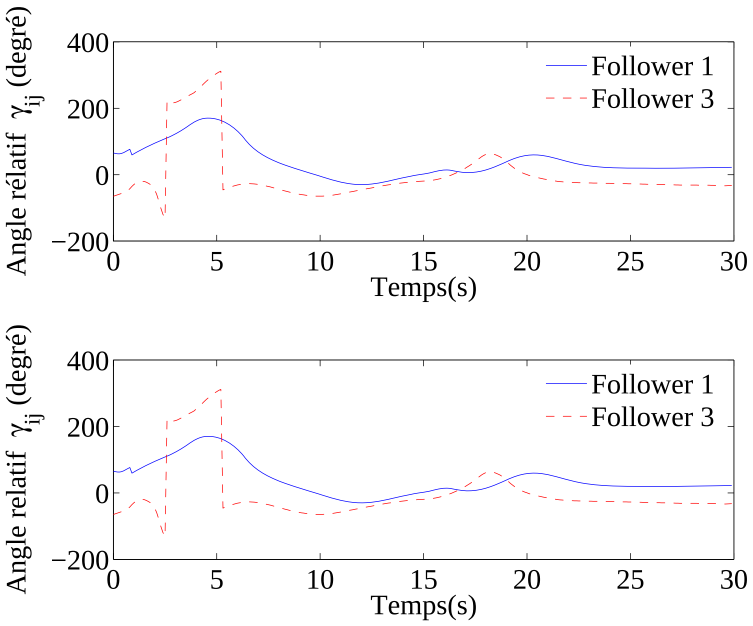


Fig. 8.9: Angle relatif γ_{ij}: $\mathscr{V} = \{1,2\}$. $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. En haut $\tau = 300ms$. En bas $\tau = 500ms$

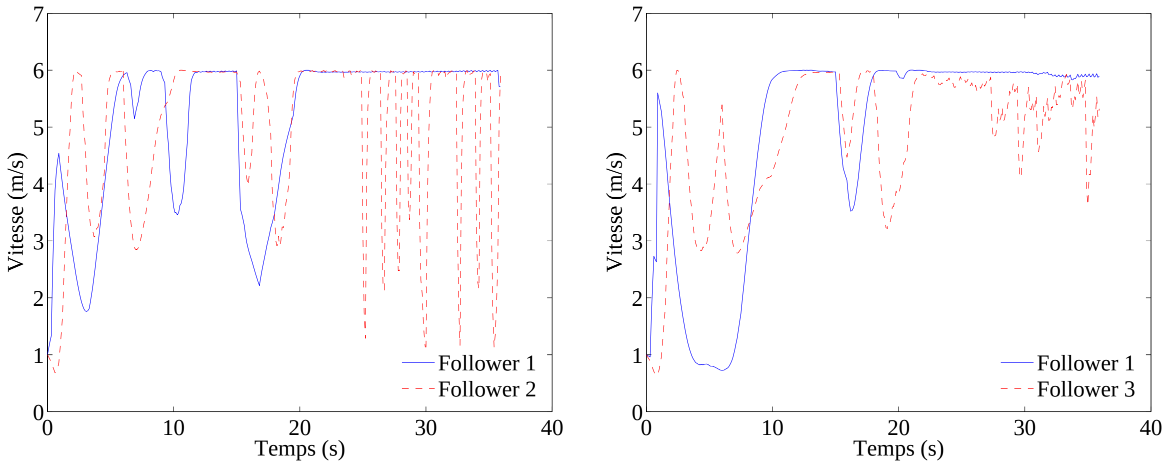


Fig. 8.10: Vitesse v_i: $\mathscr{V} = \{1,2\}$. $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Gauche $\tau = 300ms$. Droite $\tau = 500ms$

8.3 Evitement d'obstacles

Une autre contrainte que nous souhaitons inclure dans cette stratégie de commande est l'évitement de collision. Cet évitement de collision est basé sur une zone déformable entourant et protégeant le robot à partir d'informations issues de capteurs de proximité (ultra-sons) [4]. Pour un robot mobile cette zone virtuelle déformable (ZVD)(cf. figure (8.13)), sera paramétrable en fonction de la vitesse du véhicule et des connaissances éventuelles de l'environnement dans lequel il évoluera. Lorsqu'un obstacle va pénétrer dans l'environnement, une déformation va apparaître dans la zone et sera directement propagée vers le contrôleur avec pour objectif de restituer la forme initiale. On peut comparer cet algorithme à un jeu à

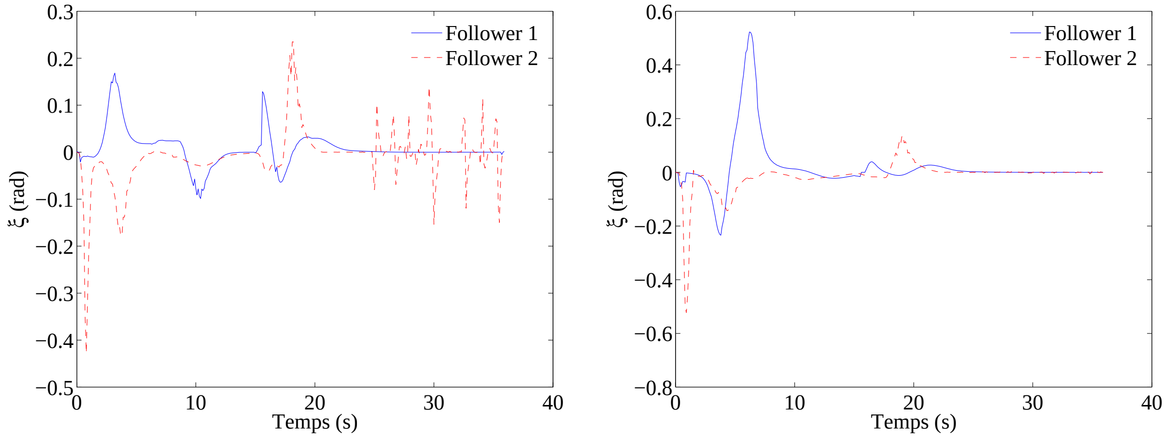


Fig. 8.11: Angle de braquage ξ_i: $\mathscr{V} = \{1,2\}$. $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. Gauche $\tau = 300ms$. Droite $\tau = 500ms$

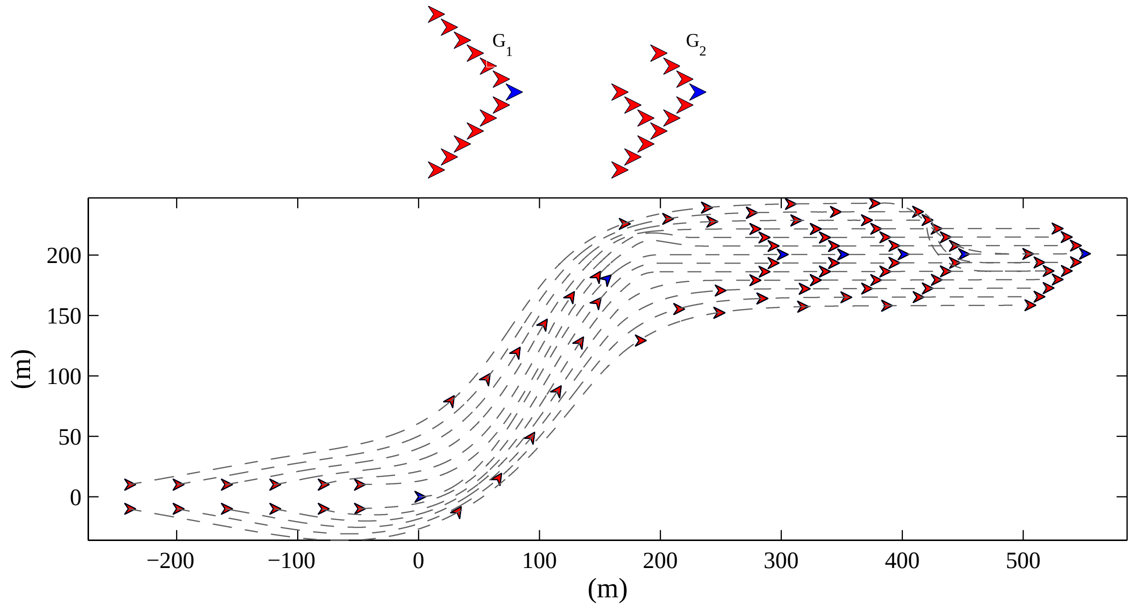


Fig. 8.12: Flotille 13 véhicules. En haut foltilles désirées, G_1 si $t \leq 11s$, G_1 si $t > 11s$. En bas, trajectoires

.

deux joueurs où le premier serait l'environnement qui créerait des déformations non désirées et le deuxième, le contrôleur du robot, qui essayerait de reconstruire la forme initiale.

Nous proposons d'incorporer la ZVD dans la fonction de coût J. En incorporant ce terme dans l'optimisation nous souhaitons obtenir une trajectoire qui minimiserait, en plus du temps, la déformation totale de la ZVD. La fonction de coût minimisant le temps et la déformation de la ZVD est définie par:

$$J(t) = \int_0^{t_f} (1 + J_{obst})dt, \tag{8.9}$$

avec J_{obst} le terme de déformation de la ZVD.

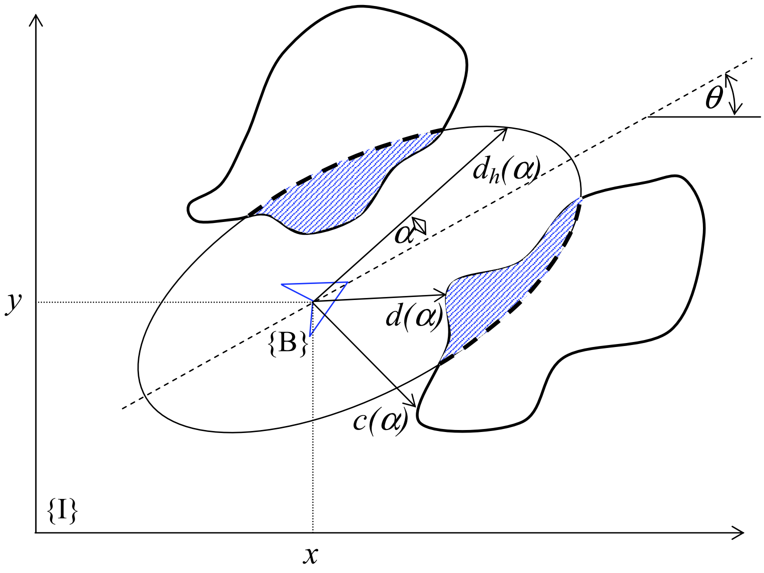


Fig. 8.13: Zone virtuelle déformable

La figure 8.14 illustre l'évitement d'obstacles pour une formation de trois robots. Lorsque les obstacles doivent être êvités la géometrie de la formation doit évoluer. Une fois les obstacles contournés la formation peut reprendre sa configuration désirée.

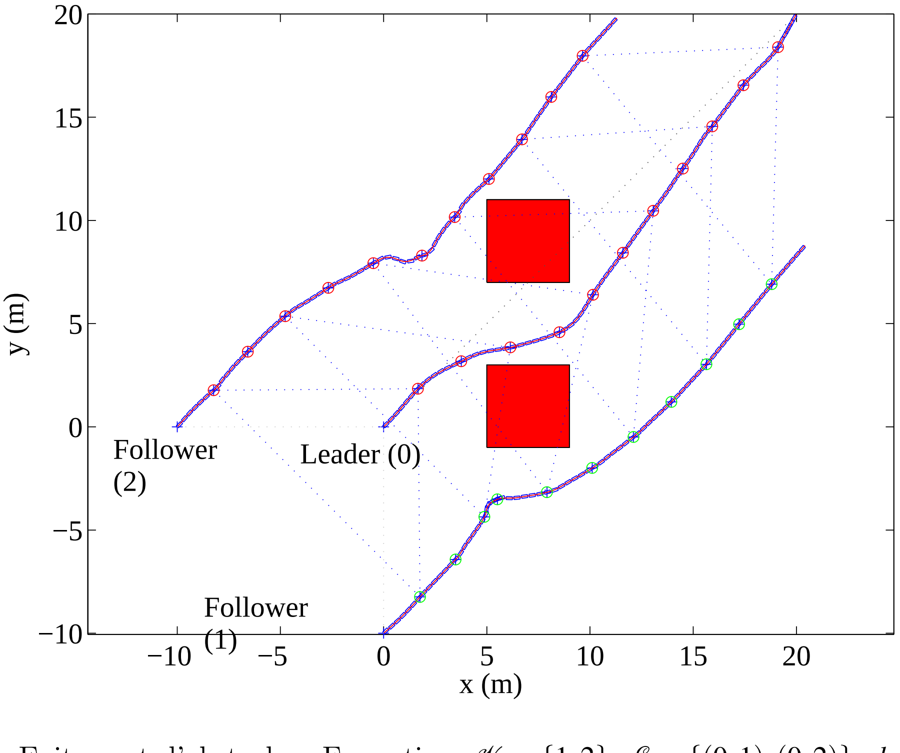


Fig. 8.14: Evitement d'obstacles. Formation: $\mathscr{V} = \{1,2\}$, $\mathscr{E} = \{(0,1),(0,2)\}$. $l_{01} = l_{02} = 10m, \gamma_{01} = -\gamma_{02} = \frac{\pi}{4}$. $T_s = 300ms$

8.4 Positionnement grâce au niveau de réception

Pour assurer le maintien du lien de communication, il est nécessaire d'introduire une nouvelle variable de commande, le niveau de réception WiFi s'exprimant en dBm. On considère que les véhicules sont munis de deux dispositifs WiFi.

Le premier grâce à une antenne omnidirectionnelle envoie à son (ou ses) follower(s) sa position courante. Le second muni d'une antenne sectorielle motorisée, est capable de suivre et de mesurer le niveau de réception de la transmission de son leader local.

Cette nouvelle variable de commande qui représente la puissance de réception peut s'exprimer en fonction de la distance entre les véhicules:

$$R_p = R_0 + 10 \cdot \alpha_{dbm} \cdot \log_{10}(l) + \nu, \tag{8.10}$$

où R_0

est la perte du signal en dBm pour une distance de 1 mètre, R_p est la perte du signal en dBm pour une distance $l > 1$ mètre, α_{dbm} est le coefficient de perte du signal, et ν une variable gaussienne répresentant les phénomènes de propagation multi-chemins associés a la propagation des signaux radios.

Nous proposons la fusion les informations données par les capteurs odométriques du véhicule et le niveau de réception du signal WiFi. Il nous faut au préalable identifier le modèle de propagation (8.10) (cf. figure (8.15). Ensuite, on estime la distance l en utilisant un filtre de Kalman étendu [68].

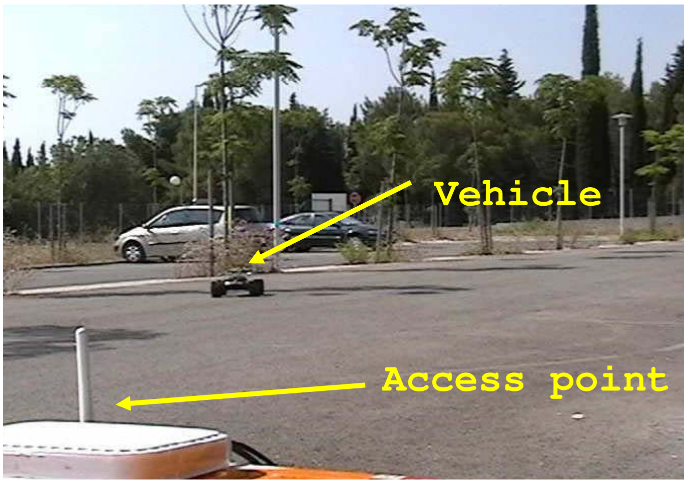


Fig. 8.15: Identification du model de propagation. Expérimentation

Le filtre de Kalman proposé pour l'estimation de la position des robots a été validé en utilisant le niveau de réception du signal WiFi (cf. figure (8.16)

8.5 Conclusions

Nous avons proposé une nouvelle méthode de génération de trajectoire réactive capable de prendre en compte un ensemble de contraintes différentes et parfois antagonistes. Les résultats que nous avons obtenus sont extrêmement encourageants.

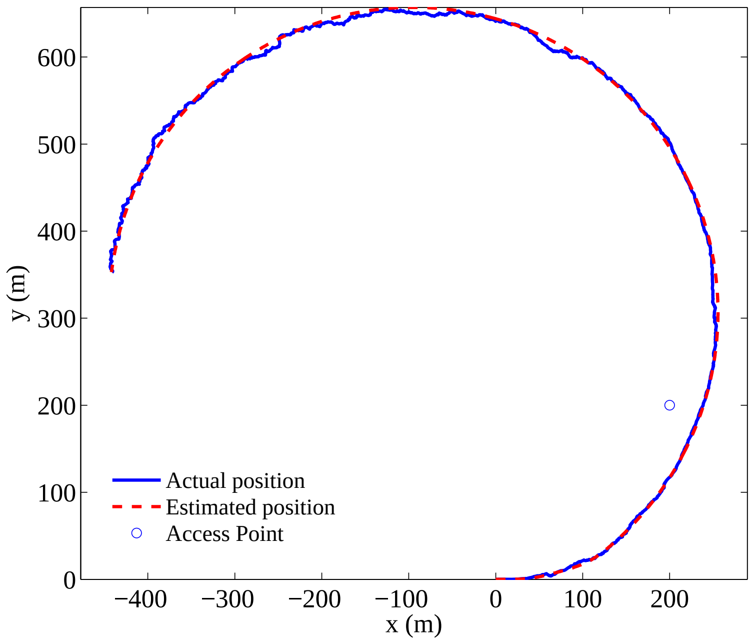


Fig. 8.16: Estimation de position pour un robot en utilisant le niveau de réception du signal WiFi

La prise en compte du niveau de réception radio en tant que variable de commande de la flotille afin de maintenir le lien de communication intact quels que soient les obstacles rencontrés, est un point important qui doit être approfondi par la prise en compte d'un modèle de rayonnement des antennes non isotrope. Une étude plus poussée de la modélisation des robots et de la commande globale comprenant les liens de communications va constituer, dans les années qui viennent, un défi scientifique important pour la commande de flottille de robots mobiles terrestres.

Sur la même voie, le domaine exploratoire que constitue la commande collaborative de robots mobiles sous marins (AUV) commence à faire l'object d'un certain interêt [109]. Même si les problèmes rencontrés en termes de commande, de génération de trajectoires ainsi que de retards de transmission sont bien plus complexes à traiter.

BIBLIOGRAPHY

[1] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 34, no. 5, pp. 2015–2028, 2004.

[2] V. Ozdemir and H. Temeltas, "Decentralized formation control using artificial potentials and virtual leaders," *6th IFAC symposium on Intelligent Autonomous Vehicles proccedings.*, 2007.

[3] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, Feb. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-4BBVX6N-2/2/00337008da152fd0c71d8df5a4dac26f

[4] R. Zapata, A. Cacitti, and P. Lepinay, "Dvz-based collision avoidance control of non-holonomic mobile manipulators," *Journal Européen des Systèmes Automatiques*, vol. 38, no. 5, pp. 559–588, 2004.

[5] R. A. Brooks and A. M. Flynn, "Fast, cheap and out of control: A robot invasion of the solar system," *Journal of The British Interplanetary Society*, vol. 42, pp. 478–485, 1989.

[6] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Structure decision method for self organising robots based on cell structures-cebot," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989, pp. 695–700 vol.2.

[7] R. Brooks, P. Maes, M. Mataric, and G. More, "Lunar base construction robots," in *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, 1990, pp. 389–392 vol.1.

[8] M. Mataric, "Minimizing complexity in controlling a mobile robot population," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 830–835 vol.1.

[9] P. Caloud, W. Choi, J.-C. Latombe, C. Le Pape, and M. Yim, "Indoor automation with many mobile robots," in *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, 1990, pp. 67–72 vol.1.

[10] F. Noreils, "Integrating multirobot coordination in a mobile-robot control system," in *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, 1990, pp. 43–49 vol.1.

[11] A. Davids, "Urban search and rescue robots: from tragedy to technology," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 17, no. 2, pp. 81–83, 2002.

[12] B. Yamauchi, "Decentralized coordination for multirobot exploration," *Robotics and Autonomous Systems*, vol. 29, no. 2-3, pp. 111–118, Nov. 1999. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-3YRNTSR-2/2/edbe1e9ba25d8b4233a173b1ad1d57f4

[13] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[14] C. Weisbin and G. Rodriguez, "Nasa robotics research for planetary surface exploration," *Robotics & Automation Magazine, IEEE*, vol. 7, no. 4, pp. 25–34, 2000.

[15] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, Apr. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2H-4M8779F-1/2/e5b43d641b91787199fd2a4ca59dd8b9

[16] D. Camacho, F. Fernandez, and M. Rodelgo, "Roboskeleton: An architecture for coordinating robot soccer agents," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 2, pp. 179–188, Mar. 2006.

[17] J.-H. Kim, D.-H. Kim, Y.-J. Kim, and K. T. Seow, *Soccer Robotics.* Springer-Verlag, 2004.

[18] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, pp. 1–23, 1997.

[19] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for swarm robots," in *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 1, 1993, pp. 441–447 vol.1.

[20] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester, "Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, pp. 842–848, 2005.

[21] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 21, no. 2, pp. 20–28, 2006.

[22] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on*, vol. 18, pp. 813–825, 2002.

[23] A. M. Bloch, *Nonholonomic Mechanics and Control.* Springer Interdisciplinary applied mathematics, 2003, vol. 24.

[24] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[25] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, pp. 367–393, 1990.

[26] J. P. Laumond, "Finding collision-free smooth trajectories for a non-holonomic mobile robot." in *10th International Joint Conference on Artifcial Intelligence*, Milan, Italy, 1987.

[27] C. Samson and K. Ait-Abderrahim, "Feedback stabilization of a nonholonomic wheeled mobile robot," in *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, 1991.

[28] R. W. Brockett, *Differential Geometry Control Theory.* Birkhauser, 1983, ch. Asymptotic stability and feedback stabilization, pp. 181–191. [Online]. Available: http://hrl.harvard.edu/publications/brockett83asymptotic.pdf

[29] A. Bloch, M. Reyhanoglu, and N. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *Automatic Control, IEEE Transactions on*, vol. 37, no. 11, pp. 1746–1757, 1992.

[30] P. Morin and C. Samson, "Trajectory tracking for non-holonomic vehicles: overview and case study," in *Proc. 4th International Workshop on Robot Motion and Control RoMoCo'04*, 2004, pp. 139–153.

[31] A. Ailon, N. Berman, and S. Arogeti, "On controllability and trajectory tracking of a kinematic vehicle model," *Automatica*, vol. 41, no. 5, pp. 889–896, May 2005. [Online]. Available: http://www.sciencedirect.com/science/article/B6V21-4FMBKCC-2/2/7e87d12361256018ff9c8024c69e5109

[32] W. Nelson and I. Cox, "Local path control for an autonomous vehicle," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, 1988, pp. 1504–1510 vol.3.

[33] M. Defoort, T. Floquet, W. Perruquetti, and A. M. Kökösy, "Tracking of a unicycle-type mobile robot using integral sliding mode control." in *2nd International Conference on Informatics in Control, Automation and Robotics*, Barcelone, Spain, 2005, pp. 106–111.

[34] J.-M. Yang and J.-H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 3, pp. 578–587, 1999.

[35] K. Park, H. Chung, and J. G. Lee, "Point stabilization of mobile robots via state-space exact feedback linearization," *Robotics and Computer-Integrated Manufacturing*, vol. 16, no. 5, pp. 353–363, Oct. 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V4P-4118JRP-5/2/ebec5a2872a162f25d007ef0a41237b2

[36] C. de Wit and O. Sørdalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *Automatic Control, IEEE Transactions on*, vol. 37, no. 11, pp. 1791–1797, 1992.

[37] O. J. Sørdalen, "Feedback control of nonholonomic mobile robots," Ph.D. dissertation, The Norwegian Institute of Technology, 1993.

[38] C. Fernandes, L. Gurvits, and Z. Li, "Near-optimal nonholonomic motion planning for a system of coupled rigid bodies," *Automatic Control, IEEE Transactions on*, vol. 39, no. 3, pp. 450–463, 1994.

[39] P. Martin, "Contribution a l'etude des systemes differenctiellment plats," Ph.D. dissertation, Ecole nationale superieure des mines de Paris, 1992, in french.

[40] M. J. V. Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *International Journal of Robust and Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, 1998. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1099-1239(199809)8:11<995::AID-RNC373>3.0.CO;2-W

[41] M. Milam, "Real-time optimal trajectory generation for constrained dynamical systems," Ph.D. dissertation, California Institute of Technology, 2003.

[42] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3, 2001, pp. 2968–2973 vol.3.

[43] J. Cruz, J., "Leader-follower strategies for multilevel systems," *Automatic Control, IEEE Transactions on*, vol. 23, no. 2, pp. 244–255, 1978.

[44] J. T. Feddema, C. Lewis, and D. A. Schoenwald, "Decentralized control of cooperative robotic vehicles: theory and application," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 852–864, 2002.

[45] D. Cruz, J. Mcclintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control - a multivehicle platform for research in networked embedded systems," *Control Systems Magazine, IEEE*, vol. 27, no. 3, pp. 58–78, 2007.

[46] X. Chen, A. Serrani, and H. Ozbay, "Control of leader-follower formations of terrestrial uav's," in *Proc. 42nd IEEE Decision and Control Conference*, vol. 1, 2003, pp. 498–503.

[47] P. J. Seiler, "Coordinated control of unmanned aerial vehicles," Ph.D. dissertation, University of California, Berkeley, 2001.

[48] B. Vanek, T. Peni, J. Bokor, and G. Balas, "Practical approach to real-time trajectory tracking of uav formations," in *American Control Conference, 2005. Proceedings of the 2005*, 2005, pp. 122–127 vol. 1.

[49] D. Edwards, T. Bean, D. Odell, and M. Anderson, "A leader-follower algorithm for multiple auv formations," in *Autonomous Underwater Vehicles, 2004 IEEE/OES*, 2004, pp. 40–46.

[50] E. Yang and D. Gu, "Nonlinear formation-keeping and mooring control of multiple autonomous underwater vehicles," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 2, pp. 164–178, 2007.

[51] N. Lechevin, C. Rabbath, and P. Sicard, "Trajectory tracking of leader-follower formations characterized by constant line-of-sight angles," *Automatica*, vol. 42, no. 12, pp. 2131–2141, Dec. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6V21-4KY88RK-1/2/e9e38cb0ddefefc4fd5887ad289fe41f

[52] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 905–908, 2001.

[53] R. Patel and F. Shadpey, *Control of Redundant Robot Manipulators*, Springer, Ed., 2005.

[54] B. E. Bishop, "Control of platoons of nonholonomic vehicles using redundant manipulator analogs," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2005, pp. 4606–4611.

[55] D. Stilwell and B. Bishop, "Platoons of underwater vehicles," *Control Systems Magazine, IEEE*, vol. 20, no. 6, pp. 45–52, 2000.

[56] V. Lumelsky and A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape." *Algorithmica*, vol. 2, pp. 403–430, 1987.

[57] I. Kamon, E. Rimon, and E. Rivlin, "Tangentbug: A range-sensor-based navigation algorithm," *The International Journal of Robotics Research*, vol. 17, pp. 934–953, 1998.

[58] J. Andrews and N. Hogan, *Control of Manufacturing Processes and Robotic Systems.* ASME, 1983, ch. Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator, pp. 243–251.

[59] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The Int. Journal of Robotics Research*, vol. 5, pp. 90–99, 1986.

[60] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California.*, vol. 1, pp. 1398–1404, 1991.

[61] S. Quinlan, "Real-time modification of collision-free paths," Ph.D. dissertation, Department of Computer Science, Stanford University, 1994.

[62] S. Quinlan and O. Khatib, "Towards real-time execution of motion tasks," 1992.

[63] F. Lamiraux, D. Bonnafous, and O. Lefebvre, “Reactive path deformation for nonholonomic mobile robots,” *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 6, pp. 967–977, 2004.

[64] H. Hu, J. Ryde, and J. Shen, *Autonomous mobile robots.* Taylor and Francis, 2006, ch. Landmarks and triangulation in navigatioin, pp. 149–184.

[65] E. Abbott and D. Powell, “Land-vehicle navigation using gps,” *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, 1999.

[66] J. Cheng, Y. Lu, E. R. Thomas, and J. A. Farrell, *Auonomous Mobile Robots.* CRC Press, 2006, ch. Data Fusion via Kalman Filter: GPS and INS, pp. 99–148.

[67] K. A. Redmill, T. Kitajima, and U. Ozguner, “Dgps/ins integrated positioning for control of automated vehicle,” in *Proc. IEEE Intelligent Transportation Systems*, vol. 1, 2001, pp. 172–178.

[68] G. Whelch and G. Bishop, “An introduction to the kalman filter,” Department of Computer Science. University of North Carolina at Chapel Hill. Course notes., 2001.

[69] P. Rives and M. Devy, *La robotique mobile.* Hermes, 2001, ch. Perception pour la localisatioin, pp. 141–198, in french.

[70] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 869–880, 1996.

[71] F. Selleri, *Relativity in Rotating Frames.* Springer, 2003, ch. Sagnac effect : end of the mystery, pp. 57–78.

[72] M. Fliess, J. Levine, P. Martin, and P. Rouchon, “Sur les systèmes linéaires différentiellement plats,” *Comptes rendus de l’Académie des sciences, Serie I*, vol. 315, pp. 619–624, 1992.

[73] L. Piegl and W. Tiller, *The NURBS book*, 2nd ed. Springer, 1997.

[74] O. von Stryk, *Optimal control theory and numerical methods.* Basel: Birkhäuser, 1993, ch. Numerical solution of optimal control problems by direct collocation, pp. 129–143.

[75] ——, “Direct and indirect methods for trajectory optimization,” *Annals of operations research*, vol. 37, pp. 357–373, 1992.

[76] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User’S Guide For Npsol 5.0: A Fortran Package For Nonlinear Programming*, 1998.

[77] P. E. Gill, W. Murray, and M. A. Saunders, *User’s Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, 2006.

[78] C. Lawrence, J. Zhou, and A. Tils, *User's Guide for CFSQP Version 2.5*, Electrical Engineering Department, University of Meryland USA, 1997.

[79] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Kluwer Academic Publishers, 2003.

[80] R. Bhattacharya, "Optragen 1.0," Aerospace Engineering Department, Texas A&M University., Tech. Rep., 2006.

[81] P. Fraisse, R. Zapata, W. Zarrad, and D. Andreu, "Remote secure decentralized control strategy for mobile robots," *Advanced Robotics*, vol. 19, pp. 1027–1040, 2005.

[82] X. Li, "Performance study of rss-based location estimation techniques for wireless sensor networks," in *Proc. IEEE Military Communication Conference*, vol. 1, 2005, pp. 1–5.

[83] J. H. Reed, K. J. Krizman, B. D. Woerner, and T. S. Rappaport, "An overview of the challenges and progress in meeting the e-911 requirement for location service," vol. 36, pp. 30–37, 1998.

[84] R. Roberts, "IEEE P802.15 ranging subcommitee final report," 2004. [Online]. Available: http://grouper.ieee.org/groups/802/15/pub/04/15-04-0581-07-004a-ranging-subcommittee-final-report.doc

[85] V. Matellán, J. M. Cañas, and O. Serrano, "Wifi localization methods for autonomous robots," *Robotica*, vol. 24, pp. 455–461, 2006.

[86] K. Pahlavan, X. Li, and J. P. Makela, "Indoor geolocation science and technology," vol. 40, pp. 112–118, 2002.

[87] Q. Yihong, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," vol. 5, pp. 672–681, 2006.

[88] P. Krishnamurthy, J. Beneat, M. Marku, and K. Pahlavan, "Modeling of the wideband indoor radio channel for geolocation applications in residential areas," in *Proc. IEEE Vehicular Technology Conference*, vol. 1, Jul 1999, pp. 175–179.

[89] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME Basic Engineering*, vol. 82, pp. 35–45, 1960.

[90] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[91] M. St-Pierre and D. Gingras, "Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system," in *Proc. IEEE Intelligent Vehicles Symposium*, 2004, pp. 831–835.

[92] S. I. Roumeliotis and G. A. Bekey, “Distributed multirobot localization,” *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 781–792, 2002.

[93] S. Rezaei and R. Sengupta, “Kalman filter based integration of DGPS and vehicle sensors for localization,” in *Proc. IEEE Mechatronics and Automation Conference*, vol. 1, 2005, pp. 455–460.

[94] K. Pahlavan, P. Krishnamurthy, and A. Beneat, “Wideband radio propagation modeling for indoor geolocation applications,” vol. 36, pp. 60–65, 1998.

[95] A. Neskovic, N. Neskovic, and G. Paunovic, “Modern approaches in modeling of mobile radio systems propagation environment,” *IEEE Communications Surveys & Tutorials*, vol. 3, no. 3, pp. 2 – 12, 3rd quarter 2000.

[96] A. Ladd, K. Bekris, A. Rudys, D. Wallach, and L. Kavraki, “On the feasibility of using wireless ethernet for indoor localization,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 555–559, 2004.

[97] T. K. Sarkar, J. Zhong, K. Kyungjung, A. Medour, and M. Salazar-Palma, “A survey of various propagation models for mobile communication,” vol. 45, no. 3, pp. 51–82, 2003.

[98] “Circular waveguide antenna for 2.45 ghz / 802.11b / wifi / wlan.” [Online]. Available: http://flakey.info/antenna/waveguide/

[99] J. Ackermann, “Robust control prevents car skidding,” *Control Systems Magazine, IEEE*, vol. 17, no. 3, pp. 23–31, 1997.

[100] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” pp. 384–389 vol.1, 1990.

[101] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. Laumond, “Stabilization of trajectories for systems with nonholonomic constraints,” *Automatic Control, IEEE Transactions on*, vol. 39, no. 1, pp. 216–222, 1994.

[102] P. Coelho and U. Nunes, “Path-following control of mobile robots in presence of uncertainties,” *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 252–261, 2005.

[103] A. Benalia, M. Djemai, and J.-P. Barbot, “Control of the kinematic car using trajectory generation and the high order sliding mode control,” in *Proc. IEEE Systems, Man and Cybernetics Conference*, vol. 3, 2003, pp. 2455–2460 vol.3.

[104] D. Chwa, “Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates,” *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 637–644, 2004.

[105] R. E. Kalman, “Contributions to the theory of optimal control,” *Boletin de la Sociedad Matematica Mexicana*, vol. 5, pp. 102–119, 1960.

[106] J.-M. Coron, “Les journées mathématiques et informatiques x-ups,” in *Les journées mathématiques et informatiques X-UPS*, 1999.

[107] T. Balch and R. Arkin, “Behavior-based formation control for multirobot teams,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[108] J. K. V. Desai, J.P.; Ostrowski, “Controlling formations of multiple mobile robots,” in *Proc. IEEE Robotics and Automation International Conference*, vol. 4, 1998, pp. 2864–2869.

[109] D. Stilwell, “Decentralized control synthesis for a platoon of autonomous vehicles,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 1, 2002, pp. 744–749 vol.1.

Appendix I

APENDIX I

I.1 PUBLICATION ACTIVITIES

I.1.1 Journals

A. Gil-Pinto, P. Fraisse & R. Zapata, A decentralized algorithm for adaptive trajectory planning for a group of mobile robots. *Robotics and Autonomous Systems.* Initial Date Submitted: 15th December, 2006.

I.1.2 International Conferences

A. Gil-Pinto, P. Fraisse & R. Zapata, Decentralized Strategy for Car-Like robot Formations, *Intelligent Robots and Systems, 2007 IEEE/RSJ International Conference on*, San Diego USA, October-November 2007.

Gil-Pinto, A.; Fraisse, P.; Zapata, R. & Dauchez, P., Decentralized stabilization strategy for car-like robot formations, *6th IFAC Symposium on Intelligent Autonomous Vehicles*, Toulouse France, September 2007.

A. Gil-Pinto, P. Fraisse & R. Zapata, Wireless reception signal strength for relative positioning in a vehicle robot formation, *3rd International Workshop on Networked Control Systems : Tolerant to Faults*, Nancy France, June 2007.

Gil-Pinto, A.; Fraisse, P. & Zapata, R. Wireless reception signal strength for relative positioning in a vehicle robot formation, *3rd IEEE Latin American Robotics Symposium*, Santiago, Chile, October 2006.

Gil-Pinto, A.; Fraisse, P. & Zapata, R. A decentralized algorithm to adaptive trajectory planning for a group of nonholonomic mobile robots, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Beijing China, 2006, 404-417.

Gil-Pinto, A.; Fraisse, P.; Zapata, R. & Andreu, D. Towards Networked Control of Robots Using Wi-Fi. *Technology Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems MTNS*, Kyoto, Japan, July 2006.

Fraisse, P.; Gil-Pinto A.; Zapata, R. & Andreu, D., Towards an adaptive trajectory planning approach for collaborative mobile robots, *1st International Workshop on Networked Control Systems : Tolerant to Faults*, Ajaccio France, October 2005.

Gil-Pinto, A.; Fraisse, P. & Zapata, R. A decentralized adaptative trajectory planning approach for a group of mobile robots. *Towards Autonomous Robotic*

Systems,London, UK, September 2005.

I.1.3 National Conferences

Gil-Pinto A., Decentralized Secure Control Strategy for Car-like Robot Formations, *Doctiss 2007*, Montpellier France, April 2007.

Fraisse, P.; Gil-Pinto A.; Zapata, R.; Divoux, T. & Perruquetti, W., *Journées Nationales de la Recherche en Robotique*, Guidel France, October 2005.

Arturo GIL PINTO

Vers une architecture de commande pour des robots mobiles coopérants non holonomes

Résumé: Ces travaux concernent l'élaboration d'une stratégie de commande décentralisée réactive pour une flottille de robots mobiles terrestres. Cette stratégie de commande est basée sur un contrôle décentralisé qui s'appuie sur le principe Leader-Follower utilisant à la fois des informations de positionnement absolu (GPS) et relatif entre véhicules (niveau de réception des liens WiFi) ainsi que des informations d'existence d'obstacles de proximités (capteurs ultra-sons). Cette méthode permet d'intégrer et d'optimiser à chaque instant ces diverses contraintes afin de générer un chemin faisable. Mais également de maintenir la flotille dans une forme géometrique donnée, avec un niveau de réception des transmissions entre les véhicules minimal, tout en évitant d'éventuels obstacles.

Mots-clefs: Commande collaborative, systèmes plats, génération de trajectoire, commande décentralisée

Towards a Control Architecture for Cooperative Nonholonomic Mobile Robots

Abstract: This work proposes a control architecture for a group of nonholonomic robotic vehicles. We present a decentralized control strategy that permits each vehicle to autonomously compute an optimal trajectory by using only locally generated information. We propose a method to incorporate reactive terms in the path planning process which adapt the trajectory of each robot, thus avoiding obstacles and maintaining communication links while it reaches the desired positions in the robot formation. We provide the proof of the reachability of the trajectory generation between the current and desired position of each follower. Simulation results validate and highlight the efficiency and relevance of this method. An integration of the wireless network signal strength data with the vehicle sensors information by means of a Kalman filter is proposed to estimate the relative position of each vehicle in a robot formation set. Vehicle sensors consist of wheel speed and steering angle, the WiFi data consist of reception signal strength (RSS) and the angle of the maximal RSS with respect to the robot orientation. A nonholonomic nonlinear model vehicle is considered; due to these nonlinearities an Extended Kalman Filter EKF is used. Simulation and experimental results of the proposed estimation strategy are presented.

Keywords: Mobile robot, Cooperative systems, Optimal control, Mobile robot motion-planning, Global positioning system, Wireless LAN.