Safe and flexible hybrid control architecture for the navigation in formation of a group of vehicles

José Miguel Vilca Ventura

N° d'ordre : 2607

EDSPIC : 712

UNIVERSITÉ BLAISE PASCAL-CLERMONT II

ÉCOLE DOCTORALE

SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

Thèse

Présentée par

JOSÉ MIGUEL VILCA VENTURA

Ingénieur Électronique

Master Robotique

pour obtenir le grade de

Docteur d'Université

Spécialité : **Vision pour la robotique**

Safe and Flexible Hybrid Control Architecture for the Navigation in Formation of a Group of Vehicles

Soutenue publiquement le 26 October 2015 devant le Jury composé de :

THIERRY FRAICHARD	Rapporteur	Chercheur à l'INRIA, Grenoble Rhône-Alpes
FAWZI NASHASHIBI	Rapporteur	Directeur de Recherche INRIA, Paris-Rocquencourt
PHILIPPE MARTINET	Examinateur	Professeur à l'École Centrale of Nantes
ANTONIOS TSOURDOS	Examinateur	Professeur à l'Université de Cranfield
LOUNIS ADOUANE	Encadrant	Maître de Conférences - HDR à l'Université Blaise Pascal
YOUCEF MEZOUAR	Directeur de thèse	Professeur à l'IFMA

ACKNOWLEDGEMENTS

The presented thesis was developed at Institut Pascal in the Modeling, Autonomy and Control in Complex system (MACCS) team of ISPR (ImageS, Perception systems and Robotics) group. I would like to express my sincere gratitude to this team for giving me the opportunity to particpate and collaborate in meaningful research trends. Without the MACCS team, this thesis would not have been possible.

I would like to thank my thesis committee: Mr. Philippe Martinet, Mr. Antonios Tsourdos, Mr. Thierry Fraichard and Mr. Fawzi Nashashibi for generously accepting to review this document. Furthermore, I would like to thank Mr. Philippe Martinet for cordially accepting to be the Chairman of the thesis committee. I wish to express my gratitude for these persons to offer their time, insightful comments and hard questions for this thesis.

I would like to thank Prof. Youcef Mezouar for directing this thesis and for his invaluable advices and encouragement during the development of this work.

Foremost, I would like to express my sincere gratitude to my adviser Prof. Lounis Adouane for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm and knowledge. His guidance helped me during all the complete process of research and thesis writing.

I also appreciate the financial support by the French National Agency (ANR) through the SafePlatoon project during my Ph.D. study.

Besides my advisers, I thank my fellow labmates, colleagues and professors at the Institut Pascal for the stimulating discussions, the unconditional support and the invaluable collaboration along the thesis. Moreover, I thank all people that I knew during my academic career for adding positives experiences in my work and life.

Last but not the least, I would like to thank my family: my parents Leonidas Vilca and Eleuteria Ventura, my brothers and sisters Diana, Jesus, Carolina, Christhian and Erick, and my nephew Mathias, for supporting me emotionally throughout my life and being my main source of motivation to achieve my goals. Thanks for everything that helped me to this day.

CONTENTS

List of Figures

List of Tables

GLOSSARY

- **AUV:** Autonomous Underwater Vehicle.
- **EKF:** Extended Kalman Filter.
- **GDI-VIPA:** Graphical Data Interface for VIPALAB.
- **GPS:** Global Positioning System.
- **MD:** Mahalanobis Distance.
- **MRF:** Multi-Robot Formation.
- **MRS:** Multi-Robot System.
- **OMWS:** Optimal Multi-criteria Waypoint Selection.
- **PAVIN:** Plate-forme d'Auvergne pour Véhicules INtelligents.
- **SFR:** Strategy for Formation Reconfiguration.
- **SMPA:** Sense-Model-Plan-Act.
- **UAV:** Unmanned Aerial Vehicle.
- **UGV:** Unmanned Ground Vehicle.
- **VIPALAB:** Véhicule Individuel Public et Autonome.

GENERAL INTRODUCTION

"Coming together is a beginning.
Keeping together is progress.
Working together is success."

Henry Ford (1863-1947)
Founder of the Ford Motor Company

Nowadays, different robotic systems are more and more present in many fields such as industries (e.g., robotic arms and manipulators), home (e.g., cleaning-robots and robotic toys), military (e.g., unmanned aerial, ground and underwater vehicles), space (e.g., robotic spacecraft and satellites), medical (e.g., robotic surgery) and more. All these systems have solved, improved or optimized the specific tasks which they were designed for. Some convincing examples of the capabilities of robotic systems are the Curiosity rover [307] for the Mars-expedition and the Philae robot for landing on a comet [376].

In recent years, one of the most interesting research corresponds to autonomous navigation of a single or a group of vehicles in structured environment. It is an important topic of research around the world due to its advantages in different domains such as ecology (reduction of pollution), transport time (efficient time schedule), safety (reduction of road accidents) and health (human comfort) among others [358, 422]. Nevertheless, many challenges remain in different domains such as in the perception in which climatological factors (rain, fog, darkness, lightness, etc.) disturb the environmental perceived data [363], as well as, in the control to design a robust and fault tolerant systems to deal with the errors and uncertainties of the sensors, in the communication in which network delays can produce erroneous decision, etc. [354].

Furthermore, the coordination of a group of robots to perform a task shows important benefits and large applications (agriculture, military or transport sector) [238, 255]. Tasks that may be performed by a single complex robot can be performed more efficiently and cheaply by a group of elementary cooperating robots. These cooperative systems can be fault tolerant (one robot can be replaced or joined for more complex tasks) [365, 385] and can improve certain criteria related to the time, the robustness or the flexibility of the task to achieve [165, 227]. Nonetheless, some difficult problems need to be solved to achieve a suitable coordination. The numerous interactions between robots themselves and their environment, as well as the complexity of task to be achieved do not permit the direct use of neither classical perception nor control techniques [388, 421].

In this context, the **Safeplatoon** project [309], to which this PhD thesis is associated, studies the problem of autonomous vehicles navigating in convoy for application in urban, military and agricultural environments. The innovative characteristics of this project is related to the design and development of the movement capabilities in a large and robust convoy. The aim of our work is the development of a control architecture for navigation in formation of a group of vehicles in structured environments.

This Multi-Robot Formation (MRF) is defined as a group of vehicles with an appropriate coordination between them in order to keep a pre-defined geometric configuration and avoiding hindering obstacles during the autonomous navigation. The strategy for the development of proposed safe and flexible control architecture is to break up the complex task (navigation in formation) in adequate elementary behaviors [260, 294]. Therefore, the desired flexibility of the control architecture can be obtained, for instance, a new controller can be added or modified to perform another complex task while keeping the initial elementary behaviors. The desired safety will be analyzed for each controller in the sense of performance, system constraints, convergence, etc. Furthermore, this control architecture is embedded in each vehicle instead of using a main central control system.

This manuscript begins with a survey of autonomous navigation for a single vehicle. The proposed control architecture has been developed for this task. The elementary controllers were designed to reach a target or successive targets while avoiding the detected obstacles. This obstacle avoidance is based on reactive methods (limit-cycle trajectories) using only the measurements of the local sensor of each vehicle [293]. This control architecture is validated in simulations and experiments. The formation control is based on the Leader-follower and behavior-based approaches where the leader sends its instantaneous pose and velocity to each follower. This approach avoids the followers' necessity of knowing *a priori* the whole leader trajectory. In addition, this control architecture is complemented with a stable formation reconfiguration (i.e., the formation shape changes), for example from line to triangle shape. This overall control architecture is validated in simulation and experiments which allows to conclude on the reliability and effectiveness of our work.

This manuscript is organized in two parts, the first part contains two chapters which correspond to the state of the art and bibliographical review:

- **Chapter 1** contains a summary of the state of the art in mobile robotics, with a main focus on autonomous ground vehicles. Different control architectures for autonomous navigation of a single robot are presented.
- **Chapter 2** contains a summary of the state of the art in cooperative robotics, with a main focus on autonomous navigation of a fleet of vehicles. Different control architectures for autonomous navigation of a group of robots are detailed. Particularly, a description of the main approaches for navigation in formation is presented.

The second part corresponds to different proposals developed during this PhD thesis. This part contains three chapters:

- **Chapter 3** is related to autonomous navigation of a single vehicle in cluttered environments. The composition of the proposed control architecture for a single robot's navigation is described. A specific block related to the environmental perception is detailed. Its role, advantages and drawbacks are also presented. Mainly, this chapter describes the proposed control law which allows to perform safe autonomous navigation. Its stability and safety are demonstrated. Simulations and experiments validate the proposed control architecture for safe navigation.
- **Chapter 4** describes the proposed navigation strategy through successive waypoints which allows to perform safe and flexible autonomous navigation. The proposed methods to select the optimal set of waypoints in structured environments

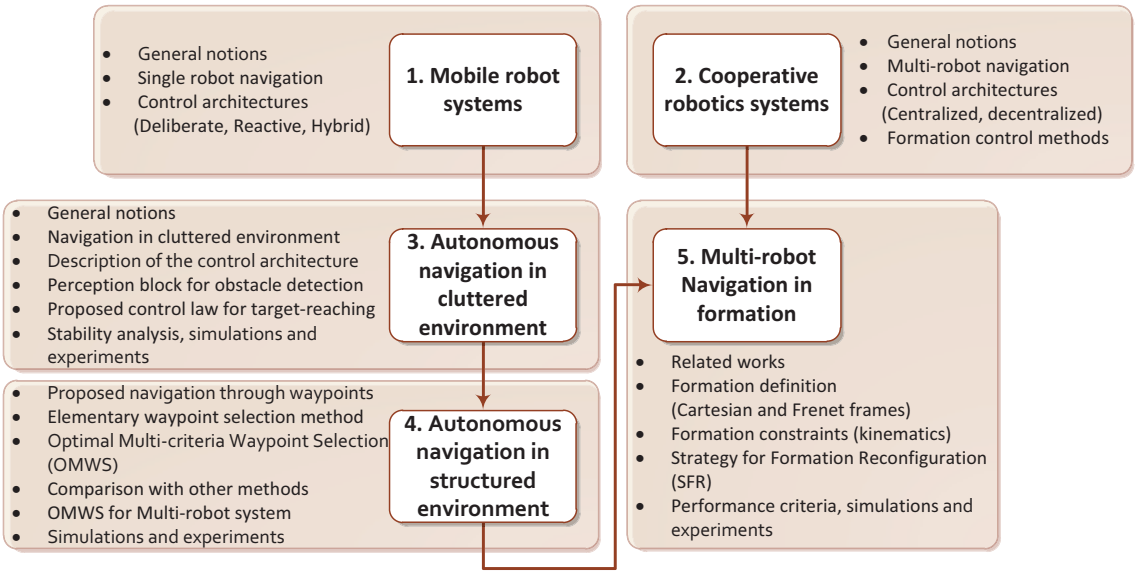


Figure 1: General schema of the manuscript.

are presented. Its advantages and drawbacks w.r.t other methods from the literature are explained. In addition, an extension to multi-robot formation is presented. Simulations and experiments are performed to demonstrate the application of the proposed strategy.

- **Chapter 5** deals with the multi-robot navigation in formation. The main literature approaches are given and our proposal is explained. Mainly, the proposed strategy for reconfiguration of the formation is presented with different examples. An exhaustive analysis of the system using different criteria is detailed in simulations and experiments.

This manuscript ends with a general conclusion summarizing the main contributions of this PhD thesis and the future works.

An overall scheme about the general organization of the manuscript is shown in Fig. 1

CONTRIBUTIONS

The goal of this work is developing a safe and flexible Multi-Robot System (MRS) control architecture for the navigation in formation of a group of vehicles. Before to design the MRS control architecture, we start by autonomous navigation of a single vehicle based on a Hybrid architecture which allows respectively a high-level planning and reactive navigation (cf. Subsection 1.3.3) [8]. Moreover, this architecture allows to add some other behaviors. Our navigation strategy for a single vehicle consists in reaching successive waypoints instead of tracking a predefined trajectory (cf. Section 4.2) [4, 9]. The idea is allowing the vehicle to perform different maneuvers between the waypoints such as obstacle avoidance before reaching another waypoints. The proposed control law is designed to converge to its assigned waypoints while guaranteeing stable and smooth vehicle's trajectories. Moreover, the advantage of this control law w.r.t others is performing different behavior such as trajectory tracking, point stabilization, dynamic target reaching [4, 8] (cf.

Subsection 3.5). The obstacle avoidance is based on reactive method (as developed in [293]) using limit-cycle. We propose an on-line obstacle detection method which allows to enclose the dangerous obstacle with an ellipse using range sensors [1, 3]. The cognitive part is related to the selection of the set of waypoint in structured environment. We propose an algorithm to obtain the optimal set of waypoint while taking into account the environmental constraints, the vehicle's kinematics and uncertainties. The advantages of this algorithm are the different considered criteria and the convergence time to obtain a solution [9].

Once the single vehicle navigation is tested and validated via simulation and experimental results, the extension to MRS is analyzed. The MRS control architecture is based on centralized/decentralized architecture. The centralized part is dedicated to the formation management, i.e., the leader sends the follower position w.r.t. it (such as the virtual structure). The decentralized part is related to the tracking of the desired pose (virtual target) by the follower. Moreover, each follower can perform different behaviors (such as obstacle avoidance [5] (cf. chapter 5)) according to the environment perception.

The advantage of this MRS architecture is to allow the dynamic formation reconfiguration (e.g., line to triangle shape, or diamond to square shape) using a virtual structure definition (cf. Section 5.2). The leader decides when a reconfiguration is necessary according to the perceived environment (e.g., when a hinder obstacle is detected). The advantages of the designed method w.r.t. others is that it takes into account the formation shape dimensions and vehicle constraints while guaranteeing the shape convergence (cf. Section 5.4). Moreover, to evaluate the performance of the MRS system, system errors, convergence time, formation shape along the navigation are presented. The proposed MRS control architecture (cf. Chapter 5) is validated via simulation and experimental results showing its safety (no collision), feasibility (easy implementation) and flexibility (add more vehicles/behavior) for a group of vehicles.

I

CONTEXT AND STATE OF THE ART

1

MOBILE ROBOTS FOR AUTONOMOUS NAVIGATION

This chapter describes the state of the art in mobile robots, particularly applied to autonomous navigation of a single vehicle. An important part of the chapter is dedicated to several control architectures of mobile robot system designed to perform autonomous navigation.

Robotics is a branch of technology such as computer science, mechanical and electronic engineering that deals with the design, construction, operation and application of different robots [425]. A robot is a machine (mechanical) capable of carrying out a complex series of actions or tasks automatically. The word *robot* was, as it is well known, introduced by the Czech writer Karel Čapek [10]. It comes from the Slavic word *robota* which means labor or work. The word *robotics* (derived from *robot*) was first used by the science fiction writer Isaac Asimov in the science fiction short history titled “Liar!” [11].

(a) Industrial robot (FANUC®) for car manufacturing [263].

(b) Quadruped robot (Boston Dynamics®) for soldier assistance in rough terrains [360].

(c) Wheelchair robot (Genny®) for transportation of disabled people [290].

(d) Suction robots (Maytronics®) for cleaning of swimming pool [381].

(e) Space robot (ESA®) for comet surface lander [376].

(f) Kangaroo robot (Festo®), Insect fly robot (Harvard) and Salamander robot (EPFL) [257].

Figure 1.1: Different robots applied to different domains.

The robotic systems have been applied to different domains, for instance: industries (car manufacturing (cf. Fig. 1.1(a))) [263, 271], military (assistance, reconnaissance, guidance and weaponry (cf. Fig. 1.1(b))) [186, 360], transportation (large heavy objects and elderly or disabled humans (cf. Fig. 1.1(c))) [290, 341], domestic (house cleaning, toys and human interaction (cf. Fig. 1.1(d))) [233, 381], space (orbiters, rovers and landers (cf. Fig. 1.1(e))) [348, 403] , and other applications as given in [239, 257] (cf. Fig. 1.1(f)).

1.1/ MOBILE ROBOT SYSTEMS

A mobile robot is an automatic machine not fixed to one physical location and capable to move in its environment [182]. Mobile robots can be classified according to their work environments as follows:

- Land robots are related to the navigation in the ground surface (e.g., desert or mountain areas) [250, 262] (cf. Fig. 1.2(a)). They are commonly called Unmanned Ground Vehicles (UGVs).
- Aerial robots are related to the navigation in the air [305, 325] (cf. Fig. 1.2(b)). They are usually named Unmanned Aerial Vehicles (UAVs).
- Underwater robot are related to the underwater navigation [287, 393] (cf. Fig. 1.2(c)). They are commonly called Autonomous Underwater Vehicles (AUVs).
- Polar robots are related to the navigation in icy areas [184, 267] (cf. Fig. 1.2(d)).

The presented work is dedicated to UGVs. The objective of these robots can vary according to the application, e.g., human or area surveillance [351, 420] (cf. Fig. 1.3(a)),

(a) Explorer robot (NASA®) named Curiosity rover.

(b) Combat aerial robot (GA-ASI®) named Predator B.

(c) Fish robot (MIT®) named RoboTuna.

(d) Polar rover robot (NASA®) named Goddard Remote Operated VEhicle (GROVER).

Figure 1.2: Different examples of mobile robots.

mapping of unknown environments [304, 355] (cf. Fig. 1.3(b)), human search and rescue [333, 405] (cf. Fig. 1.3(c)), space exploration [230, 289] (cf. Fig. 1.3(d)), autonomous navigation [219, 330] among others [273, 415]. Furthermore, the UGV can use different devices to move such as legs (bio-inspired robot) (cf. Fig. 1.1(b) and 1.3(d)) [346, 417] wheels (car-like robot) (cf. Fig. 1.1(c) and 1.3(b)) [353, 383] and tracks (trunk-like robot) (cf. Fig. 1.2(a) and 1.3(c)) [181, 409].

In this thesis, wheeled robots, referred as UGVs, will be used with final main objective of autonomous navigation in formation. The proposed navigation strategy is divided into two phases:

(a) Navigation of a single UGV: Safe and flexible navigation has been developed. This single robot must navigate in cluttered and structured environments while avoiding the detected obstacles (cf. Chapters 3 and 4).

(b) Navigation in formation: The proposed navigation is extended to a group of UGVs, of which the leader performs a safe navigation and the other robots follow virtual dynamic targets defined according to the leader configuration. The proposed formation reconfiguration acts when one UGV detects a possible collision of the formation with any dangerous obstacle (cf. Chapter 5).

The next sections provide a brief review of the different works of autonomous navigation for a single UGV (cf. Section 1.2).

(a) Rovio for Wi-fi enabled surveillance developed by WowWee®.

(b) Landshark UGV (Black-i Robotics) with complex mapping sensor system developed by SRI®.

(c) Battlefield Extraction-Assist Robot (BEAR) for search and rescue developed by Vecna Technologies®.

(d) Eight-legged walking robot (Scorpion) for space exploration developed by University of Bremen - Germany.

Figure 1.3: Different applications of UGVs.

1.2/ SINGLE ROBOT NAVIGATION: OVERVIEW

A brief overview of the evolution of the UGV's structures is detailed in Fig. 1.4. The first mobile robots has been built by W. Grey Walter in 1949. They were turtle-like robots with three wheeled, light sensor, touch sensor, drive and steering motors and two vacuum tube analog computer [12, 90]. He named them Elsie and Elmer turtles (ELectro MEchanical Robots, Light Sensitive) (cf. Fig. 1.4). These robots (Machina Speculatrix) used one vacuum tube to imitate two interconnected neurons. These simple amplifier circuits connected the light and touch sensors to two motors. The first sensor (photocell) was connected to the drive and steering motors. The second sensor (contact switch) indicates that the turtle's shell had collided with an object. Consequently, the sensor sent the vacuum-tube amplifiers into oscillation and changed the robot's direction. This system allowed the turtles to wander in a room and return to a recharging station.

In the 1980s, the largest robot car project in the field of driverless car, named Eureka PROMETHEUS Project (PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety), was launched with the cooperation of several European companies such as Ernst Dickmanns' team of UniBW, Daimler-Benz, Jaguar, BMW and others [26, 36, 80]. The final Mercedes prototypes, the autonomous W140 S-Class called VaMP (*Versuchsfahrzeug für autonome Mobilität Passenger* which means "test vehicle for autonomous mobility Passenger") and VITA-2 (VIsion Technology Application) (cf. Fig. 1.5(a)), used saccadic computer vision, sixty transputers (a type of parallel computers), and probabilistic approaches (Kalman filter) to react in real time to road and traffic conditions [63, 68, 87, 102] (cf. Fig. 1.4). The steering, throttle and brakes were controlled through computer commands based on a real time processing of image sequences taken by four cameras with different focal lengths for each hemisphere. This system allowed to track other vehicles and read road markings automatically.

In the 1990s, National Aeronautics and Space Administration (NASA) launched the Mars Pathfinder mission, with the objective of landing of a first robotic rover (Sojourner) to the Mars surface [201, 245] (cf. Fig. 1.4). Sojourner was built with six-wheel instead of four-wheel since it allows more stability and obstacle-crossing capabilities [103, 115]. The control commands were operated from Earth using a graphical interface of Rover Control Software. The software allowed to Sojourner to capture its surrounding from various angles, carrying the analysis of terrain features or locating waypoints [93, 94].

In the 2000s, DARPA [424] (Defense Advanced Research Projects Agency) and ELROB [427] (European Land-Robot Trial) challenges motivated the development of new vehicle technologies for different complex environments in military, rural and urban areas [215, 220, 241, 264, 338]. In the last years, the development of fully autonomous vehicle for transportation field has received more attention in different countries [352]. The objective of autonomous vehicles is to improve quality of life, as pollution reduction and accidents prevention, etc. [354, 358, 422]. One of the UGVs that has drawn the public's attention is the Google driverless car [285, 313] (cf. Fig. 1.5(b)). This vehicle has a LIDAR mounted on the top which allows a view of $360°$. This data is used to generate a high-resolution 3D map of its environment (accurate to about $11\ cm$). Currently, the system needs to have a prior map of the area (pre-defined map) where the vehicle is supposed to drive, including traffics lights, telephone poles, crosswalks and static infrastructure [313]. This map allows to quickly identify dynamic objects (e.g., pedestrians and cyclists) and to improve the accuracy of the localization system based on GPS, gyro-

scopes, altimeters and tachymeters. This information is merged by the on-board software to analyze the traffic situation and generate a safe vehicle's path. The on-board software

Figure 1.4: Brief overview of UGV's developments.

(a) VaMP car developed by Prometheus project.

(b) Google driverless car developed by Google.

Figure 1.5: Examples of driverless cars.

system controls brake, gas and steering systems. Other vehicle manufacturers developing the autonomous vehicle are Mercedes (Mercedes F015), Audi (Audi A7) and BMW (BMW M235i) among others [343, 422].

This thesis focuses on the autonomous navigation task (level 4/5 according to [359, 394] (cf. Fig. 1.6), i.e., driverless vehicle which performs a safe navigation while solving different critical situations). This task is defined as the displacement of a single and multiple UGVs from a started point to an end point (goal or target) in a structured area (with static and dynamic obstacles, roads, corners, roundabouts and others).

The following section describes the main control architectures allowing autonomous navigation of UGVs.

BASt level	NHTSA level	SAE level	Narrative definition		Execution of steering and acceleration/ deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)
Human driver monitors the driving environment								
0	Driver only	0	No Automation	The full-time performance by the human driver of all aspects of the dynamic driving tasks, even when enhanced by warning or intervention systems.	Human driver	Human driver	Human driver	n/a
1	Assisted	1	Driver Assistance	The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task.	Human driver and system	Human driver	Human driver	Some driving modes
2	Partially automated	2	Driver Assistance	The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task.	**System**	Human driver	Human driver	Some driving modes
Automated system ("system") monitors the driving environment								
3	Highly automated	3	Conditional Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with expectation that the human driver will respond appropriately to a request to intervene.	System	**System**	Human driver	Some driving modes
3/4	Fully automated	4	High Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene.	System	System	**System**	Some driving modes
		5	Full Automation	the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver.	System	System	System	**All driving modes**

Figure 1.6: Summary of the levels of driving automation for on-Road vehicles [394].

1.3/ CONTROL ARCHITECTURES FOR UGVS

Autonomous control of an elementary single UGV basically consists of four main processes (cf. Fig. 1.7) which are described below:

(a) Perception: Different sensors are used to obtain an accurate data from the UGV and its environment:

- Proprioceptive sensor: It obtains information about the current robot's state (position, orientation, velocity, battery level, etc.). Some examples are: odometers, gyroscope and Inertial Measurement Unit (IMU).
- Exteroceptive sensor: It obtains information from the environment. Some examples are Global Positioning System (GPS), camera, range sensors (LIDAR, sonar and infrared).

One of the task of the perception system is to detect different obstacles (static or dynamic) in the environment. It is a crucial process since the localization and decision process depend upon it. This process must provide a good estimation of the obstacle positions in the near surroundings of the vehicle and this with low latency and a high update frequency. The main approaches use range sensor [81, 401] and cameras [297, 410]. The low-cost of the camera has motivated many works in this area such as stereo-vision, omnidirectional camera, monocular, etc. [83, 367, 390]. The range sensors have an accurate measurement in different environmental conditions (fog, rain, etc.) [286, 362]. This thesis dedicates a short part for this process using range sensors (LIDAR) (cf. Section 3.4).

(b) Localization: This process determines the vehicle's pose w.r.t. its work environment. It is a critical process since most of other processes depend upon it. The main localization methods are based on proprioceptive sensors (odometers [160, 274] and gyroscope [70, 288]) and exteroceptive sensors (GPS [117, 178],

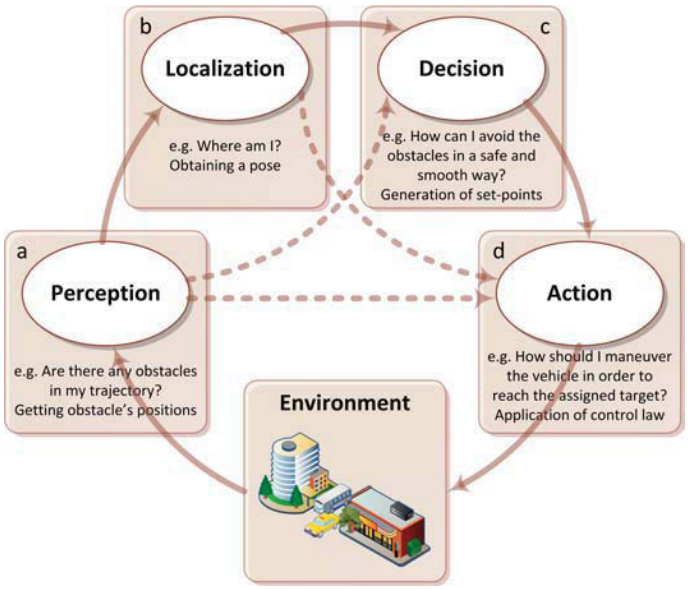


Figure 1.7: Reference control scheme for UGV's autonomous navigation.

camera [171, 205], range sensors [20, 148]). Commonly, a data fusion from the mentioned sensors is required to provide an accurate estimation of the vehicles' pose. A great extent of research focus on the data fusion using Extended Kalman Filter (EKF) [138], Markov localization [105], particle filter [268] or robust filter dealing with data uncertainties [252]. Nevertheless, some drawbacks are still existing in real application in which some elements can influence the localization such as luminosity for camera or material type for range sensor or indoor environment for GPS, as well as the time processing for data fusion of several sensors, and accuracy of sensors [247, 284]. Moreover, different approaches from literature have been developed to contribute to *Perception* and *Localization* processes such as Simultaneous Localization and Mapping (SLAM) [192, 246, 304, 349]. SLAM can be defined as the problem of building a map of an unknown environment by a mobile robot while navigating at the same time in the environment. The main challenging issue of SLAM is that a good map is needed for accurate localization while an accurate pose estimation is needed to build a map [182, 192]. This thesis focuses in the development of the control system. We assume an accurate localization system allowing to perform all simulation results. In experiments, a data fusion using EKF with RTK-GPS and gyrometer is implemented to obtain an accurate pose.

(c) Decision: This process is the outcome based on the different elements such as the assigned task, the perceived environment from perception process, the vehicle's pose and the priority of the system. For instance, the vehicle navigates towards its goal, an obstacle is detected, therefore, the decision process has to choose if the vehicle stops or the vehicle must maneuver to avoid it according to the task to accomplish. Different architectures are designed to take into account the decision elements such as LAAS architecture (LAAS Architecture for Autonomous Systems) [98, 131], which is organized in three hierarchical levels named decision, execution and functional levels. The decision level includes the task planner and is also supervising its execution. The execution level receives tasks which transforms into procedures composed of elementary robot actions and supervises its execution while being reactive to the environment. The functional level embeds a set of elementary robot functions implementing control loops (motion control, perception, etc.).

Another example is CLARAty architecture (Coupled Layer Architecture for Robotic Autonomy) [139, 166] which consists of two distinct layers named Functional and Decision Layer. The Functional layer defines the various actions of the system and adapts the action components to real devices. The Decision layer is in charge of high-level autonomy using the information from the Functional layer, the global resources and mission constraints. Furthermore, other architectures are based on artificial intelligence such as multi-agent systems where the interaction between virtual agents and the sensor data provides a merged command for execution [6], [374]. In this work, the decision process is an elementary process where the non-collision of the vehicle with any obstacle is the highest priority (cf. Chapter 3).

(d) Action: This process applies the commands generated by the decision process to the robot. The control law generates the commands to apply to the actuators (motor) according to the sensor information, decision process (maneuvers) and vehicle model. This command law can be very complex according to the system modeling and the desired task. For instance in navigation application, the common design of the command law is oriented to guide the vehicle along a trajectory [194, 318]. Moreover the control law can take into account the uncertainties of the sensors and

actuators (robust control [252, 356]), and can even deal with fault in the system (fault-tolerant control [344, 411]). In this thesis, the designed control law is used for navigation while taking into account the UGV's modeling and its constraints (cf. Chapter 3).

In the sequel, three kind of control architectures, Deliberative, Reactive and Hybrid architectures, for autonomous navigation are described according to the process detailed above (cf. Fig. 1.7) [188, 244, 294].

1.3.1/ DELIBERATIVE ARCHITECTURES

This kind of architecture is also named Sense-Model-Plan-Act (SMPA). It is characterized by sending the perceived information to several processing stages (Decision process) based on modeling and planning methods to obtain a consistent plan for current situation. Then, the plan is successively decomposed by other function modules until the desired actions can be directly executed by the actuators. The straightforward way to depict that is horizontal (or lateral) decomposition (cf. Fig. 1.8). This SMPA architecture using three functional elements: sensing (camera), planning and executing systems was implemented in Shakey robot of Stanford University [15]. Another direct implementation of the SMPA architecture is the NASREM architecture [27, 41].

The Deliberative approach for autonomous navigation contains localization and decision modules (cf. Fig. 1.8). This approach assumes a complete or quasi-complete world modeling between sensing and acting [14, 15]. This world modeling can be used for localization w.r.t. a map of the environment using the collected sensor data. The world modeling requires more effort to be built (it requires accurate sensors and fast processing time) allowing an architecture that can successfully navigate in different environments.

The drawbacks of this approach is to design an accurate and reliable world modeling for internal representation of the robot. If the model diverges from reality (i.e., if the map is wrong), then the robot's navigation may be dangerous and incorrect due to possible misguided commands. Another drawback of this approach is that the several stages, through which all sensor information must be processed, can generate an inevitable high delay in the loop between input and action [14, 15]. This delay can be reduced by improving the

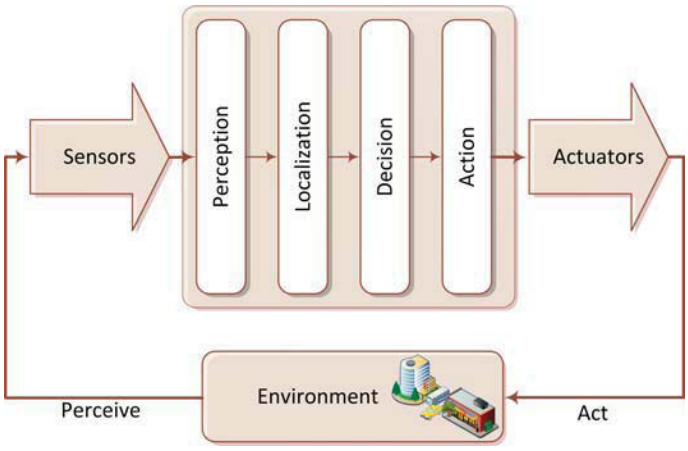


Figure 1.8: Example of Deliberative or SMPA architecture.

processing capabilities of the modules.

1.3.2/ REACTIVE ARCHITECTURES

This approach is based on stimuli-response, e.g., the control system reacts to its sensors and drives its actuators. It was proposed in opposition to the Deliberative architectures to avoid to rely upon world modeling. This approach uses minimal or any state information which implies that the behaviors may not directly accomplish the goal since the robot could not have any feedback related to current state of its assigned task (cf. Fig. 1.9). Reactive approach was inspired by biological systems which have evolved patterns of behavior without requiring neither exact models for control nor detailed planning before they can act [100].

Main advantage of this approach correspond to its very simple computational implementation which ensures fast response to external events enabling the robot's navigation in very dynamic environments [100, 188, 244]. Different approaches have been developed to create reactive systems with diverse levels of abstraction . An example is Subsumption architecture developed by Brooks [17]. Each module is an instance of a fairly simple computational machine named Behaviors. This approach proposes a vertical decomposition and the sensor data is used to select actions in an intimate and bottom-up way (cf. Fig. 1.9). Other examples are behavioral architecture [152, 377] and some approaches based on neural network [169] or fuzzy logic [154].

Main disadvantages of reactive architecture are the lack of state information, in particular world models. This lack of information makes it difficult to enable look ahead planning and goal-directed behavior (i.e., the robot has to reached a specific position named goal or target). Another drawback is the elementary behaviors that have to be thoughtfully designed to produce the desired actions. This behavior/process design may be time-consuming and might heavily rely on the specific robot hardware and its environment [239]. Moreover, this approach can have several active behaviors at any time which produce several switches between multiple behaviors damaging the robot's hardware. The addition of each new behavior forces the robot designer to reanalyze the whole existing architecture to ensure its accurate overall stability [260].

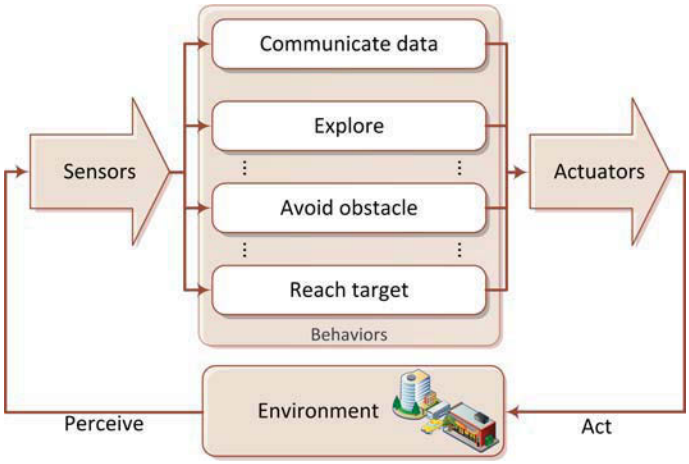


Figure 1.9: Example of Reactive or Subsumption architecture.

1.3.3/ COMPARISON AND COMBINATION OF ARCHITECTURES

One of the most important discussion in the domain of control systems for autonomous robots is the Deliberative versus Reactive architectures [167]. One problem of Reactive approach is the extremely difficult design of the primitive behaviors for all situations even for unforeseen situations [53]. As the higher-level behaviors are designed with a knowledge of the characteristics of the lower-level ones, a modification to a primitive behavior may produce modifications to several other behaviors. Furthermore, the modeling of intelligence based on behavior does not scale to human-like problems and performance [79]. For instance, the ability to search images to find stimuli on which to act is a necessary component of any intelligent agent. Another drawback of reactive approach is that the robot does not have an action plan, its global behavior is constructed from the interactions between the limited behaviors which may be insufficient to perform more complex behaviors since these complex behavior can require other behaviors which are not embedded in the robot [73].

Reactive approaches may have more limited abilities since most of the strictly Reactive robots are not based on complex sensors. Nevertheless, the deliberative architectures based on artificial intelligence (AI) may not suitable for real-time application in dynamic world [43].

The main problem of the selection of a kind of architecture is to find the right equilibrium between the desired intelligent behavior and the necessary reactivity, since:

- The existence of a specialized agent deliberating about the access to the environment usually improves the quality of the robot's response to changes according to its environment. However, it can also introduce time delays between the decision and the execution of the actions.
- A central module with a global world modeling generally leads the system to make better choices about the actions to accomplish, however it is a potential communication bottleneck.

Therefore, the selection of the appropriate architecture is a technological problem for which the designer must consider the required degrees of reactivity and intelligent behavior, efficiency to perform the task and the related implementation costs [67]. For instance, if a reactive approach is used, the reactivity of the robot can be easily obtained, however some work is necessary to ensure that the current robot behavior corresponds to the desired action in all situations. On the other hand, with a Deliberative approach the robot responses are more predictable, but an excessive complexity of the policy of access to the environment model can dangerously increase its reaction time.

Several architectures, named hybrid architectures have been designed to overcome the drawbacks related to traditional deliberative and reactive architectures, i.e. world modeling, intelligence and reactive mechanisms.

Table 1.1: Comparison between architectures

Deliberation	Reactive
Slow response velocity	Fast response velocity
Cognition	Reaction
World modeling	No modeling

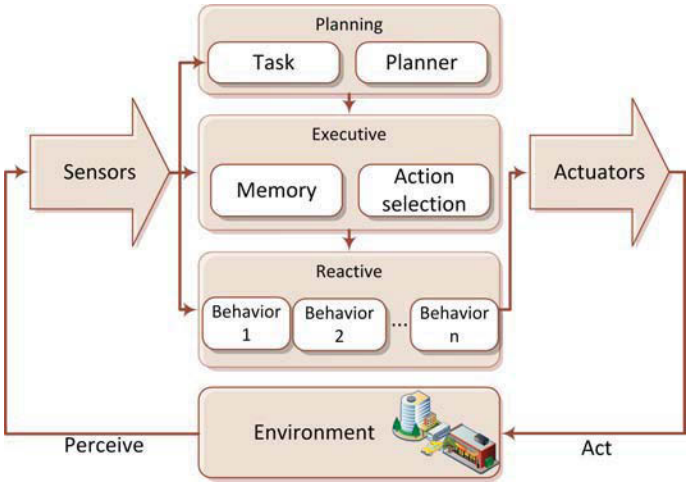


Figure 1.10: Example of Hybrid architecture.

Three layer architectures are developed to integrate both architectures [98, 131]. These layers uses three levels of abstraction, a planning, an executive and a reactive layers (cf. Fig. 1.10). They are described below:

- The Planning layer correspond to strategic planning as well as eventual plan modification. This level relies on very abstract knowledge, highly complex reasoning methods and the typical domain of AI planners.
- The Executive layer correspond to reactive planner which selects and executes appropriate strategies with regards to the designed rules. A strategy is an ordered set of actions, i.e. hierarchies of actions. Execution of strategy leads finally to activation/deactivation of reactive layer behaviors.
- The Reactive layer performs the transition from planning to numerical control and the combination of separate behaviors.

An advantage of the hybrid approach is the elimination of some levels of hierarchy according to the considered application. For instance, an implementation without the higher planning level (only decisional and functional levels), since the control architecture is addressed to short planning based on pre-written ordered sets of actions is presented in in [96]. Moreover, the authors demonstrate the communication and interaction between the modules applied to the real world. The LAAS architecture is an example of a hybrid architectures [98, 131]. The different elements to sense (perception modules), model (mapping modules), plan (planners) and act (actuator modules) indicates the presence of deliberative approach. The hierarchy of levels is implemented in vertical decomposition. The modules, under certain circumstances, can operate in parallel to directly enter predefined sensors and actuators, indicating the presence of a reactive approach. Other examples are Hughes Research laboratory software for planning and control requirements of an autonomous land vehicle [19, 47], Blackboard architecture [16], control architecture for CaRINA Intelligent Robotic Car [378], SOAR architecture [269], Task-Control Architecture (TCA) [53, 67] and others [167].

Different applications, works and control architecture of UGVs are shown in the previous sections. The following chapter extends this concept for a group of UGVs whose cooperate to achieve the assigned task.

1.4/ CONCLUSION

This chapter provided an overall review of mobile robots research with a special focus on autonomous navigation field. Due to the large amount of works, only some relevant works for single navigation with its architecture are described. This description aims to present related works and justify the choice of the proposed architecture. The described control architecture were classified according to the decision capacity of their components, the reactive system keeps the sensor and/or actuators connected to all modules, and the deliberate system requires a processing and a management of a data information by a planner. The navigation of a single vehicle is addressed (cf. Section 1.2), the proposed architecture allows reactive navigation using behaviors and local information. The cognitive part allows the planning of the navigation through the optimal set of waypoints instead of using any trajectory planning. The proposed architecture benefits of the main advantages of the deliberative and reactive architectures described in Section 1.3.

2

COOPERATIVE MULTI-ROBOT NAVIGATION

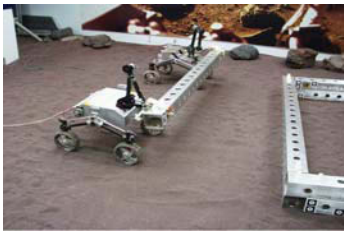
This chapter describes the state of the art in mobile multi-robot systems with a particular focus on autonomous navigation of a group of vehicles. An important part of the chapter is dedicated to several control architectures of multi-robot system designed to perform navigation in formation. Various formation control approaches are also detailed.

The basis of the cooperative robotics can be defined by “Given some task specified by a designer which may be performed by a single complex robot, a group of elementary robots displays cooperative behavior if, due to join their capacities and information, there is an increase in the total utility of the system such as more robustness, more flexibility and more efficiency w.r.t. the performance of a single complex robot” [88].

(a) Autonomous humanoid robots playing soccer in Robot Soccer World Cup (RoboCup®) [364].

(b) Cooperative quadrocopters catching a ball by ETHZ - Switzerland [336].

(c) Cooperative manipulators for spot welding by KUKA® systems [279].



(d) Two cooperating rovers carrying long container by NASA® [170].

(e) Group of quadracopters building a tower by GRASP lab (University of Pennsylvania-USA) [328].

(f) Heterogeneous small autonomous robots dynamically connected by Swarmanoid project [322].

Figure 2.1: Different examples of multi-robot system.

These cooperative robotic systems can be classified in two categories [244]:

- **Collective swarm systems**: Each cooperative robot executes its assigned task with minimal information related to the other robots. Some multi-robot application fields are exploration in hazardous environments [272, 329, 371], entertainment [364, 366] (cf. Fig. 2.1(a)), mapping of unknown locations [237, 384, 387], management and platooning of autonomous vehicles [89, 210, 323] (cf. Fig. 2.4), rendezvous of multiple agents [223, 226, 368] and coverage of unknown area [296].
- **Intentionally cooperative systems**: Each cooperative robot has information related to the other robot, and the global task is performed according to their current configurations and/or abilities. Some multi-robot application fields are search and rescue [112, 336, 397] (cf. Fig. 2.1(b)), cooperative manipulation [242, 279, 312] (cf. Fig. 2.1(c)), transport of heavy objects [118, 170, 175] (cf. Fig. 2.1(d)), construction [328, 408] and reconfigurable robots [322, 406, 412].

Over the last decades, research in cooperative robotics has been developed (cf. Fig. 2.2), mainly for its advantages w.r.t. single robot system such as:

- Achievement of the task by a group of simple robots instead of a single complex robot [225, 396] since the task is distributed among robots [340, 423].
- Parallelism solutions of the group of robots instead of a sequence of solution of a single robot [91, 140].
- Robustness of the group of robots (one can be changed or added related to a redundant system) [197, 418]. Moreover, each robot has an elementary design and construction instead of a single powerful well-equipped robot [386, 391].

2.1/ OVERVIEW

A brief overview of the evolution of the multi-robot systems is detailed in Fig. 2.2. In the 1970s, Distributed Artificial Intelligence (DAI), in contrast to earlier AI, emerged to deal with the coordination and interaction problems involving multiple intelligent software agents [84, 370]. The two main approaches are multi-agent systems and distributed problem solving [370].

The rapid progress of research in cooperative robotics has begun in the 1980s mainly in the field of mobile robots [88, 122, 244]. Commonly, some projects were developed in Cellular (or reconfigurable) robot systems (CEBOT project [23, 24] and SWARM project [22, 55]), in Self-organization of agents [32], in MRS architectures (ACTRESS project [30]) and in MRS motion planning (traffic control [39], formations [28, 33, 50] and GOFER project [35]).

In the 1990s, the fast development of reliable communication systems have increased largely different work in MRS (cf. Fig. 2.2). Many works were based on behavior-based approaches [43] and implemented in physical cooperative robots [72, 75]. In 1994, the European PROMETHEUS project [26, 36, 80] tested the twin robot vehicles VaMP and VITA-2 driving more than $1000\ km$ on a Paris multi-lane highway in standard heavy traffic at maximum velocity of $130\ km/h$ (cf. Fig. 2.2). They demonstrated the use of vision-based control for autonomous driving in free lanes, convoy driving, automatic tracking of

other vehicles and lane changes with autonomous passing other cars [44, 95, 130, 295]. In 1997, California's Partners for Advanced Transit and Highways (PATH) project [419] demonstrated multiple automobiles (Honda-PATH vehicles) capable of platoon driving and safe maneuvers such as lane changes [109, 110] (cf. Fig. 2.3(a)). The communication system is used to send the required information to the controllers of each vehicle (lateral (steering) and longitudinal (spacing and speed) controllers) [48, 49].

In the 2000s, the cooperative systems were extended to cooperative control of multiple UAVs which has became a more attractive research area around the world [125, 174, 180, 199]. The multi-robot systems inspired by biological systems [193] for large number of robots such as swarm intelligence [149], flocking and schooling [151, 162] were more developed. In 2004, the Centibots project funded by DARPA [202, 218] demonstrated the use of 100 robots (Amigobots and Pioneer 2 AT from ActivMedia Robotic) for mapping,

Figure 2.2: Brief background of Multi-robot systems.

(a) Convoy of a group of automated cars by PATH project.

(b) Platoon of a group of automated trucks by PATH project.

Figure 2.3: Platoon of a group of automated vehicles.

tracking and surveillance of an urban area. This project exploited the existing research in collaborative multilayer architecture and distributed robotics to apply on very large group of robots [218] (cf. Fig. 2.2).

In the last years, some of the most important research in cooperative robotics has been focused on the urban transportation system [372, 379, 414], cooperative air and ground vehicles for military applications [326, 373, 399] and mobile sensor networks [375, 407]. In 2011, a group of automated trucks was driven in platoon in PATH's project [317, 419] (cf. Fig. 2.3(b)). The experiment used three tractor-trailer trucks to demonstrate the energy savings in constant velocity and different maneuvers (to join and split from the platoon) and traveling up and down grades [314, 361]. The trucks were coupled electronically using vehicle to vehicle (V2V) communication and forward sensors to keep a specific inter-vehicle distance $(10\ m)$ [314, 361]. In 2012, Google presented a fleet of seven autonomous Toyota Prius hybrids driving more than $224000\ km$ in California State [258, 306, 400]. The vehicles combine data from Google Street View with cameras and LIDAR to obtain the vehicle's localization and to navigate safely in urban environments [258, 306].

Nevertheless, some of the specific challenges in MRS are still existing such as [238, 395, 404]:

- Accurate evaluation criteria for the performance of the multi-robot system.
- Scaling to larger numbers of robots (e.g., thousands) for cooperation tasks.
- Extensions to heterogeneous multi-robots (i.e., coordination of underwater, ground and aerial robots).
- Developing online replanning to deal with highly dynamic environment.
- Creating and implementing correct interaction approaches between the robots and human-robot.
- Design and incorporating robot motion and sensor constraints.
- Integrating the developed theoretical methods in real robots.
- Uncertainty, delay and fault-tolerant communication among the robots.

This thesis deals with the problem of MRS navigating in formation (i.e., each UGV of the group keeps a predefined position relative to each other or relative to a reference), mainly, applied to a group of UGVs for urban (private vehicles [303, 392]) , agriculture (tractors

and farm equipments [380, 389]) and military (fleet of military vehicle [129, 190]) fields. In this context, a brief review of different projects around the world on platooning of MRS are presented in [298, 392] such as:

- **SARTRE** project (SAfe Road TRains for the Environment) for automated close-formation vehicle platoons (Volvo car and trucks) in public traffic environments [280, 320]. This project was funded by European Commission; from September 2009 to September 2012.

- **CityMobil** project for public transportation of automated transport shuttles Cybercars [195, 203] driving a low velocity in restricted well-known environments with few pedestrian [335, 342]. This project was funded by the European Commission; from May 2006 to December 2011.

- **Connect & Drive** project for vehicle platoon (Toyota Prius vehicles) testing the cooperative Adaptive Cruise Control (ACC) system [282, 303] which use an algorithm designed to dampen traffic shock waves to decide whether the vehicles may accelerate, decelerate, or maintain their current velocity [270, 308]. This project was funded by High Tech Automotive System (HTAS), Dutch Ministry of Economic Affairs, Agriculture and Innovation, and ING Car Lease; from January 2009 to Mars 2011.

- **SafePlatoon** project [309, 347] studies the problem of autonomous vehicle navigating in convoy for application for urban, military and agriculture environments (cf. Fig. 2.4). The innovative characteristics lie in the design and development of the movement capabilities in an extensive and robust convoy. The project takes into account several geometric convoy configurations (linear,triangular, front line, etc.). It also integrates the ability of dynamic adaptation of the convoy configuration. An important aspect of the SafePlatoon project lies in the fact that the decision algorithms and proposed control/command will be verified and validated in actual experiment. This project was funded by Agence National de Recherche (ANR) - France , from Mars 2011 to September 2014.

Next section shows the main control architectures used for MRS navigation in formation.

Figure 2.4: Different vehicles used in Safeplatoon project.

2.2/ MULTI-ROBOT CONTROL ARCHITECTURES

Concerning the navigation in formation, the MRS architecture has the same essential elements as a single robot architecture, perception, localization, decision and action (cf. Subsection 1.3). In addition, MRS architecture has **Communication** process. This process deals with the following questions: “What do I have to do in the formation?”. Furthermore, an architecture has to include elements dealing with the robot interaction and group behavior to achieve the navigation in formation.

The design of the MRS architecture determines the properties of the system such as robustness, flexibility, fault-tolerant, scalability, etc. [122, 189, 244].

In the sequel, the most common approaches for the MRS architecture based on management of the control (to centralize the control in a single robot/entity or to decentralize the control among the robots [188, 244, 294]) are described below.

2.2.1/ CENTRALIZED ARCHITECTURES

In this architecture, a single central unit of control (one supervisor robot [54] or entity [106]) coordinates the whole group of robot to execute the assigned task. This centralized controller has the necessary sensors to observe the robots and their environment, then it can compute the set-point (pose and velocities) to send to each robot to accomplish the assigned global task [99]. One advantage is that the centralized controller can easily broadcast a command message (e.g. emergency stop) for all robots.

The factors of motivation for this architecture are [106]:

- Use of elementary robots, e.g., the robots have basic elements to achieve the task (wheels, motors, communication) and the perception sensors and processing data are made by the centralized controller.
- The centralization of the sensor information allows a better interpretation of the environments which can improve the decision process instead of rely on only the local perception of each robot (without merging process).

Main characteristics of this architecture are a high time-consuming in term of computation and processing of the data with a suitable frequency of communication of the whole system with the data (very important for real-time) [54, 106, 209]. Main drawbacks are the good performance of the centralized controller even when the group of robots are larger and the vulnerability of one controller in the case of fails or system errors.

An example of centralized architecture is proposed in [209]. This architecture named RoboSkeleton architecture (cf. Fig. 2.5) is based on the three layer architecture (cf. subsection 1.3.3) and it contains an agent that manages the other agents (cf. Fig. 2.5). If the controlling agent (CoachAgent) fails, the whole system could readily fail.

2.2.2/ DECENTRALIZED ARCHITECTURES

This architecture is characterized by the distribution of controller resources among the group of robots (e.g., perception sensor, processing unit, etc.) [144, 159]. This characteristic is required for multi-robot teams in which each robot takes decision based on

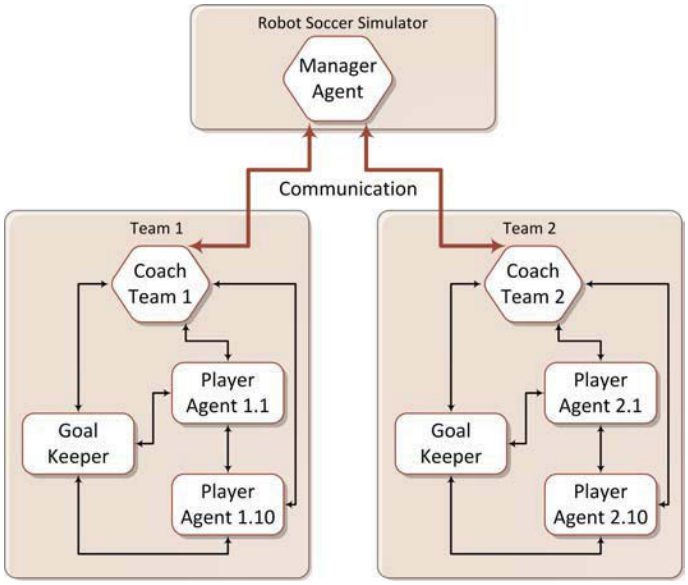


Figure 2.5: Example of centralized architecture: RoboSkeleton [209].

its local information and communicates the progress to the others of their assigned task [157, 217]. There are two types of decentralized architectures [88, 244]:

- Distributed architecture: all robots have the same control [207, 256]. Some examples are ACTRESS (ACtor-based Robots and Equipments Synthetic System) [30, 34], SWARM systems [55, 64, 76], GOFER [35, 38] and ALLIANCE [66, 108].

- Hierarchical architecture: this architecture is locally centralized [96, 152], i.e., the robots performs independently its assigned tasks, however, they send to a central entity information about the state of tasks. Some examples are CEBOT (CEllular roBOTics System) architecture [24, 60, 74], CAMPOUT (Control Architecture for Multirobot Planetary OUTposts) [135, 161] and UM-PRS (University of Michigan Procedural Reasoning System) [65, 120]

The advantages of this decentralized architecture are the robustness to system errors and failure (since no robot commands or depends on any other robot), possible parallel solution, reliable and scalable implementation [88, 144, 145, 256]. Moreover, some complex tasks which require parallel processing can be solved using this architecture [126]. The drawbacks are a hard level of coordination since the assigned tasks of each robot are embedded in the local control and if any change in the task occurs then a global reconfiguration in the robots's tasks may be difficult without a supervisor.

A typical example is ALLIANCE/L-ALLIANCE architecture developed by Parker [66, 85, 108] (cf. Fig. 2.6) for heterogeneous robots. Robots have information about their actions and the actions of the group through the top-broad communication topology. This architecture uses behavior-based controllers which depend on the assigned robot's task (motivational behaviors). L-ALLIANCE [85] is an extension of ALLIANCE which uses reinforcement learning to adjust the activation parameters of the behaviors.

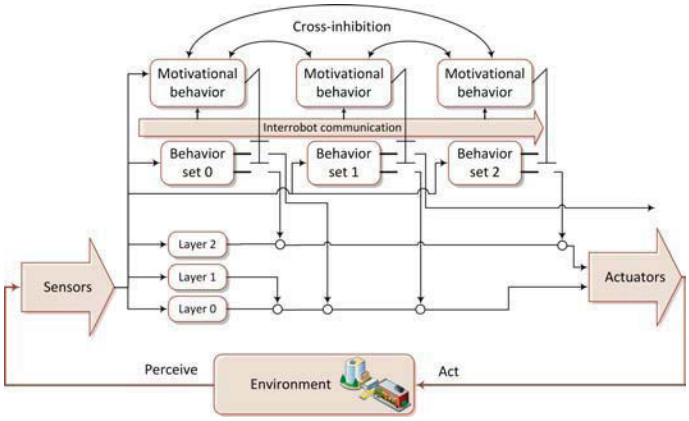


Figure 2.6: Example of decentralized architecture: ALLIANCE.

2.2.3/ CENTRALIZED/DECENTRALIZED ARCHITECTURES

The Centralized/Decentralized architecture combines the high level controller with the local control of each robot (Centralized/Decentralized [58, 244]). Therefore, the main advantages of both architecture are used such as central planner as high-level control over the autonomous robots, the robustness to errors, reprogramming of the global task/control, elimination of some hierarchical levels [96].

One example of Centralized/Decentralized architecture is DIRA (DIstributed Robot Architecture) or Layered Multirobot Architecture (LMA) developed by Simmons [136] (cf. Fig. 2.7). It is based on the three layer architecture (cf. subsection 1.3.3) and it allows coordination between robots (layer communication) and autonomy action of each one (behavior and sequencing layer). Other examples in the literature that use this architecture are Syndicate architecture [198], Emotion-Based Control architecture [150] and Multi-agent control architecture [156].

This thesis uses the Centralized/Decentralized architecture since it is more suitable for our application: navigation in formation. The centralized approach allows to the central entity (Leader) to manage the configuration of the desired formation even for formation reconfiguration. The decentralized approach allows to each UGV (follower) locally generate its commands to track its assigned virtual target. This virtual target is related to the received leader's information according to the formation shape. The following section exposes the different approaches to keep a specific formation shape.

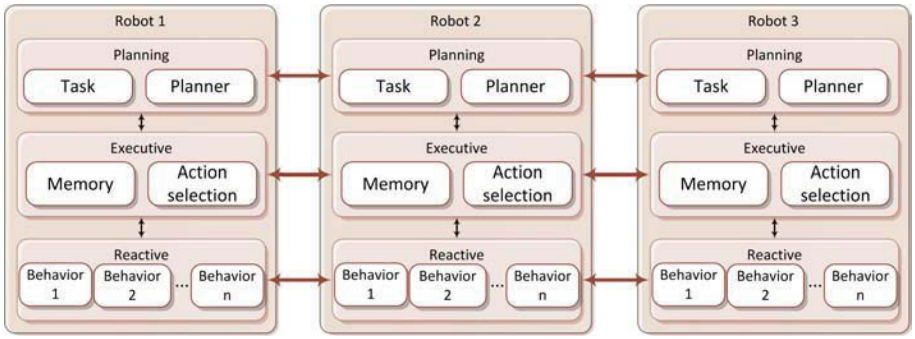


Figure 2.7: Example of Centralized/Decentralized architecture: DIRA or LMA.

Table 2.1: Comparison between MRS architectures [188].

	Perception	Localization	Decision	Action
Centralized	A central entity uses the information from all sensors to localize the robots in the environment and to compute the commands to sent to them.			Each robot execute the command sent by the central entity.
Decentralized	Each robot uses local information from its sensors to localize itself and to generate and execute its commands to accomplish the assigned task.			

2.3/ FORMATION CONTROL

Many research laboratories have been focused in navigation in formation applications, as the current thesis work, since they can be used in transport, agricultural and military fields [303, 392] (cf. Subsection 2.1).

The formation control is the ability to keep the position of a group of UGVs relative to each other or relative to a reference, i.e., the group of robots will maintain a desired geometric formation [40, 129, 238]. The main existing complications in the navigation in formation of a group of UGV include:

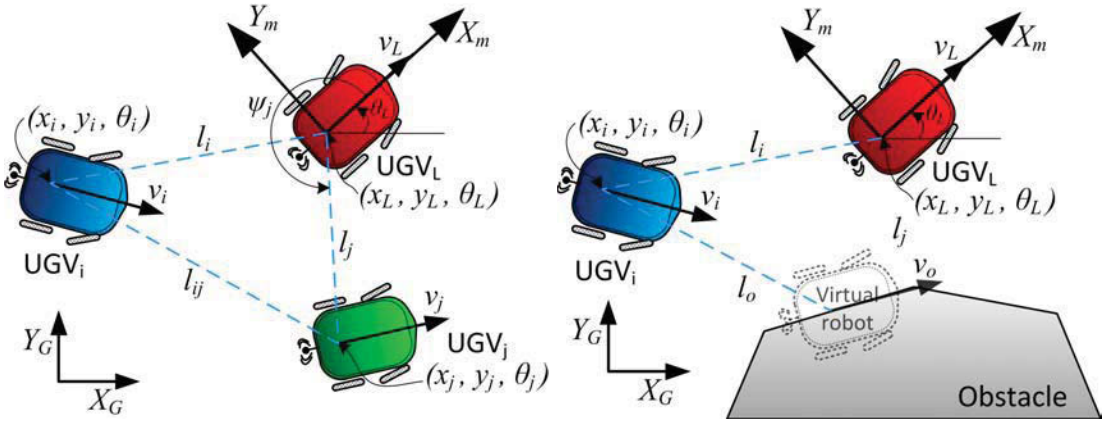
- What is the desired formation?
- How do robots to determine their desired position in the formation?
- How do robots to determine their actual position in the formation?
- How do robots to move to ensure that the formation is maintained?
- What should robots do if there are obstacles?
- How do the robot formation performance is evaluated?

Different works have been developed to reply to these questions using different approaches such as Leader-follower, Virtual Structure, Behavior-based, Potential fields and Optimization-based [191, 238] which are described below.

2.3.1/ LEADER-FOLLOWER APPROACH

The formation is defined by one robot (leader) and the others (followers) define their positions according to the leader or another guide robot using their local information from their sensor (camera, LIDAR, etc.) [61, 142, 323]. The followers can use three different aspects of information to keep the desired formation shape [104, 191]:

- Distance and orientation control $(l-\psi)$: The follower has to keep a certain distance l_j and orientation ψ_j from the leader or guide robot (cf. the UGV$_j$ in Fig. 2.8(a)).
- Distance and distance control $(l-l)$: The follower has to keep certain distances w.r.t. two other or more robots (cf. the UGV$_i$ in Fig. 2.8(a)). This $(l-l)$ control is more flexible than $(l-\psi)$ since $(l-l)$ control can use as reference one robot with another



(a) $l-\psi$ (UGV$_j$) and $l-l$ (UGV$_i$) controls. (b) $l-l$ (UGV$_i$) control for obstacle avoidance.

Figure 2.8: Leader-follower formation.

one or even with an object. For instance, this control can be extended to the case of obstacle avoidance where a virtual robot (in the obstacle border) can be used to keep a specific distance to the obstacle (cf. UGV_i in Fig. 2.8(b)).

The advantages of this approach are the implementation using decentralized architecture where each robot takes its own decision according to its local information [61, 142, 144]. Some works use the control graph theory to represent the formation interaction or communication [113, 142, 158, 210, 311]. This theory allows to prove the suitable performance and convergence of the formation under certain conditions [144, 185]. The formation shape is maintained according to the links between the robots which can vary dynamically according to robot interaction constraints, e.g., fault communication, no perception of the leader, etc. [128, 315]. In [142], three non-holonomic robots with omnidirectional cameras for vision-based control were used for different formation applications such as scouting, object transportation and formation reconfiguration (from triangle to line shape when an obstacle is detected). Different formation cases (leader reassignment, robot adding and control saturation) were presented in [113]. The authors proposed a formation control law based on the combination of Linear Matrix Inequalities (LMI) and hybrid system.

The Leader-follower approach in column formation (one UGV (Leader or follower) always visible for the followers) is described in [302]. The followers track the leader's trajectory by using only local information (camera or laser). Another representation of the Leader-follower approach is based on a spring-damper system (cf. Fig. 2.9) [235, 323]. This method proposes using two springs interaction model which allows easy implementation and transition between formation shape according to the attachment points of both springs (e.g., echelon (formation in which the UGVs are positioned diagonally) to column).

The main drawbacks of the Leader-follower approach are the dependency on the Leader (such as in the centralized architecture), e.g., any problem in the leader can stop the navigation or can produce a collision among the robots. Sometimes, the leader can not supervise the state of all followers, therefore any fault in the followers are not managed by the leader in the navigation in formation [294].

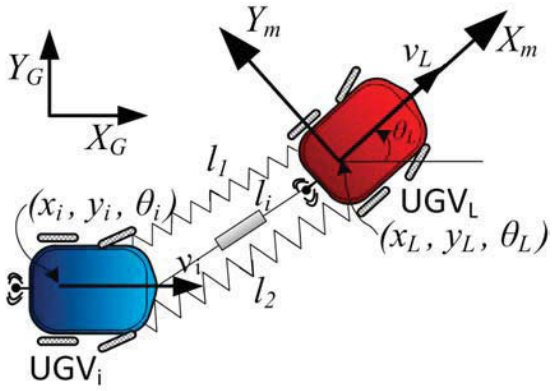


Figure 2.9: Leader-follower based on spring-damper system.

2.3.2/ VIRTUAL STRUCTURE APPROACH

This approach defines a virtual structure or rigid body, in which the group of robots in formation is considered as a single rigid body with its dynamic properties [86, 208, 369]. This approach may be derived in three steps [191]:

- The dimension, shape and dynamic of the virtual rigid body are defined according to the number and to the robots' characteristics.
- The motions of each virtual agent or node are computed according to the desired motion of the rigid body (commonly, a predefined trajectory is generated for the virtual structure).
- Each robot tracks its assigned agent/node.

The advantages of the virtual structure approach are relative simple formation definition (robots follows the center of the body instead of another robot), highly flexible for the different kinds of formation shapes and robust of the robots [248, 294, 321]. Different works use this approach, mainly applied to spacecraft or small satellite formation flying control [116, 168]. The desired formation is achieved by having all members of the formation tracking assigned nodes of the virtual structure which moves into desired configuration through a pre-defined reference trajectory. In [124], the virtual structure method is combined with a leader-following method and behavioral approach (cf. Subsection 2.3.3) to formation control of multiple spacecraft interferometer in the deep space. A similar idea was applied for spacecraft formation flying control in [168]. Each spacecraft uses a synchronized control law based on tracking of the desired state of the virtual structure to achieve the desired formation patterns. In [231], the virtual structure is used to define the followed trajectory. The dynamic model of the robot is used to design the nonlinear formation control to the trajectory synchronized tracking.

The drawbacks of the virtual structure, as the centralized approaches, is the central control unit to define and supervise the formation keeping (high time consuming due to predictive computation of the system in a finite time) [173]. Moreover, the constraints of the rigid structure according to the dimension of the vehicle and dynamic structure have to be considered.

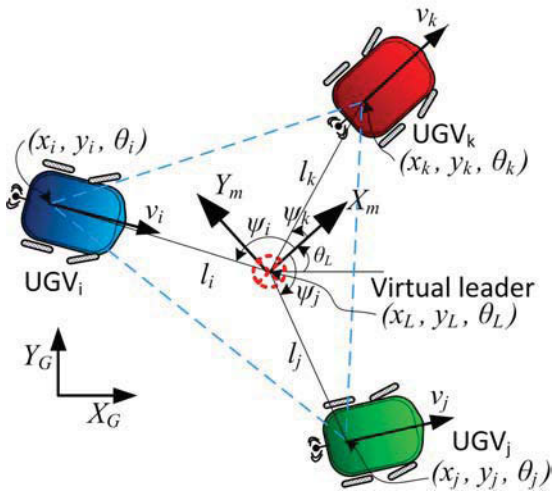


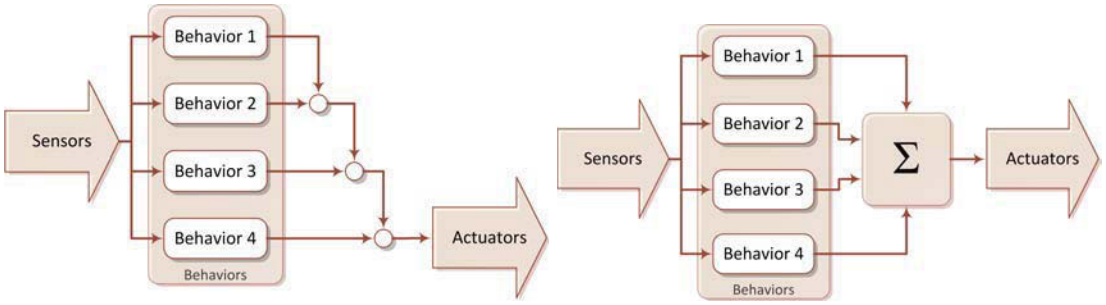
Figure 2.10: Virtual structure definition.

2.3.3/ BEHAVIOR-BASED APPROACH

This approach is based on elemental behaviors (controllers) in order to generate the desired emergent behavior [100, 101, 193] (cf. Subsection 1.3.2). The global task is decomposed in different elemental behaviors which can be selected or processed to generate the desired action according to the following action selection methods:

- **Competitive selection**: One behavior is activated according to the perception system (cf. Fig. 2.11(a)). The design is based on hierarchy in which one behavior has the highest priority w.r.t. the others. One of the drawbacks is the stability analysis during the switching phase [78, 100].

- **Cooperative selection**: Two or more behaviors are activated and the action (commands) are fused (mean, artificial intelligent fuzzy logic, multi-agent system) to obtain a final command to the robot [208] (cf. Fig. 2.11(b)). One drawback is the partial accomplishment on the assigned task due to command fusion. Some works deal with this problem [100, 276]. Moreover, as in the last case, the stability analysis (transition phase) is a very complex problem [78, 100].



(a) Competitive behavior-based. (b) Cooperative behavior-based.

Figure 2.11: Behavior-based formation control.

The advantages of the behavior-based approach are the flexibility, extension to more complex tasks and easy implementation in real robots. Different works using behavior-based approach in real robot are found in the literature. In [56, 101], the authors use a cooperative selection in which the local control rules are defined by the designer. In [146], the robots use only local information to keep a certain distance and orientation from another robot (friend) within its its field of view. The authors show several formation shapes such as diamond, triangle, arrowhead, wedge and hexagon. The problem of keeping four marching robots in a side-by-side formation is presented in [59]. The authors use a cooperative selection for merging global and local information which allows to perform obstacle avoidance or other maneuvers. In [208], the collision avoidance is taken into account by the behavior-based approach. The authors propose a Null-Space-Based (NSB) behavioral control (defined as competitive-cooperative approach) to coordinate the formation while performing different tasks. The null space of an elementary behavior is the state space region where the whole system can operate without affecting the actions of this elementary behavior. This approach is based on task-priority kinematic control approach for ground fixed manipulators to manage the possible conflicts between tasks [21].

2.3.4/ OTHER APPROACHES OF FORMATION CONTROL

2.3.4.1/ POTENTIAL FIELDS APPROACH

This approach uses virtual bodies to model the group of robots and formation dynamic. The potential fields of these bodies depends on robot role (leader or follower), robot dynamics (velocity), distance between the robots (repulsive force if vehicles are very close and attractive force if vehicles are far enough within certain distance) and distance to the obstacles (repulsive force) [153, 179]. The asymptotic stability of the formation is guaranteed [179]. Commonly, a “virtual leader” is defined to guide the movements of the group of robots [133, 134, 179]. In [153], the potential fields is defined according to the desired formation while considering its dimension constraints. The decentralized implementation allows each vehicle of the formation to perform a collision-free movements. In [196], the authors present artificial potential trenches which represent the formation structure by queues and formation vertices, instead of nodes. The artificial potential trench (associated with a queue) attracts the robot to the bottom of the valley created by the potential trench. This approach improves the scalability and flexibility of robot formations for formation reconfiguration and obstacles avoidance. One drawback of this Potential fields approach is the local minima problem and the modeling of the robot and the environment (obstacles) [45, 191].

2.3.4.2/ OPTIMIZATION-BASED APPROACH

This approach is based on optimization method using a suitable cost function [141, 204]. This cost function takes into account the error between a vehicle and its neighbors, distance to the obstacle, desired velocity, and other constraints (penalty function) [176, 321]. The optimization problem can be solved using either centralized [321] or a distributed methods [212]. Commonly, these methods uses receding horizon Model Predictive Control (MPC) [121]. The MPC method rely on an optimization of predicted model response with respect to plant input to determine the best input changes for a given state. The

MPC method is suitable for nonlinear and time-variant models with hard constraints (that cannot be violated) or soft constraints (that can be violated but with some penalty) [213]. In [176, 212], the problem is distributed for each robot which obtains the previous optimal control trajectory at each horizon time. In [321], the authors use non-linear MPC as penalty function (distance to the virtual target, obstacle) and a priority strategy (according to the distance to the object) to obtain a safe navigation of formation of a group of UAV. A simple dual-mode MPC algorithm using local leader-follower controller was presented in [57, 114].

The drawbacks of the Optimization-based approach are the time consuming due to predictive computation w.r.t. the horizon time which can originate delay in the system output, the adequate selection of the horizon time to obtain fast and suitable solution, and the limited region of convergence of the system to obtain stable and feasible solutions.

In this thesis, the formation problem is addressed by the combination of the main advantages of different approaches such as Leader-follower, virtual structure and behavior-based to obtain complex formation shapes which can be extended or reconfigured dynamically according to the environment context/configuration.

2.4/ CONCLUSION

This chapter provided an overall review in the navigation in formation field. As in the case of single robot navigation, only some works among the huge number of research are described. The common MRS architecture are presented to motivate the proposed MRS control architecture based on centralized/decentralized architecture. The proposed control architecture uses the centralized approach to allow the central entity (Leader) to manage the configuration of the desired formation even for formation reconfiguration. The decentralized approach allows to each UGV (follower) to locally generate its commands to track its assigned virtual target. Moreover, the different approaches of formation control have been detailed. The proposed formation control uses a combination of Leader-follower and behavior-based to obtain complex navigation in formation.

II

AUTONOMOUS NAVIGATION

3

SAFE HYBRID CONTROL ARCHITECTURE FOR REACTIVE AUTONOMOUS VEHICLE'S NAVIGATION IN CLUTTERED ENVIRONMENTS

Fully autonomous vehicle's navigation is a complex problem of major interest for the research community. Systems capable of performing efficient and robust autonomous navigation are unquestionably useful in many robotic applications such as manufacturing technologies [310], urban transportation [254], assistance to disabled or elderly people [331] and surveillance [155]. Even if a lot of progress has been made, some specific technologies have to be improved for real applications. This chapter is particularly focused on the problem of the autonomous navigation for UGVs in a cluttered environment (cf. Fig. 3.1).

An important issue for successful UGV's navigation is the obstacle avoidance. This function permits to prevent UGV's collision and to ensure UGV's safety. One part of the literature considers that the robot's control is entirely based on path planning methods while involving the total knowledge of its environment. All obstacles configurations are taken thus into account in the planning step. Some examples are Voronoï diagrams and visibility graphs [46] or artificial potential fields functions [52]. Moreover, the planning methods can also deal with dynamic environment while regularly replanning the robot's path [111, 200]. Nevertheless, planning and replanning require a significant computational time and complexity.

Another part of the community is concentrated on reactive methods to deal with the obstacle avoidance, where only local sensors information is used rather than a prior knowledge of the environment [143, 243, 260]. In [18], the author proposes a real-time obstacle avoidance approach based on the principle of artificial potential fields. In this work, it is assumed that the robot actions are guided by the sum of attractive and repulsive fields. In [29], the author extends Khatib's approach while proposing specific schema motors for mobile robots navigation. Another interesting approach, based on a reflex behavior reaction, uses the Deformable Virtual Zone (DVZ) concept, in which a robot movement depends on risk zone surrounding the robot [187]. If an obstacle is detected, it will deform the DVZ and the approach consists of minimizing this deformation by modifying the control vector. This method deals with any obstacle shape, however, it suffers as schema motors from local minima problem. In general, reactive methods do not require high com-

putational complexities since robot's actions must be given in real-time according to the perception [100].

Several works proposes reactive methods to deal with the obstacle avoidance problem such as obstacles avoidance using vortex fields [62] and orbital trajectories [163]. The last approach is build on circular limit-cycle differential equations in [163, 216, 261]. The circular limit-cycles are more stable than the vortex fields and always converge to periodic orbit (cf. Appendix E.3). In [293], the authors extend this limit-cycles method to elliptical trajectories. Therefore, more generic and efficient obstacle avoidance is performed for different obstacle shapes even for long walls. Indeed, an ellipse fits better this kind of obstacles than a circle (cf. Fig. 3.2).

This thesis exploits the adaptive elliptical limit-cycle method (cf. Subsection 3.2) for reactive obstacle avoidance using only robot's pose information and uncertain range data. This method was selected because it is stable and robust while using only local information from range sensors. It considers that each obstacle is surrounded by an ellipse. The ellipse parameters are computed on-line using the sequence of uncertain range data (cf. Section 3.4).

The problem of obstacle avoidance and the considered scenario during the UGV's navigation is presented in the next section. The proposed control architecture to deal with this problem is described in Section 3.3. Sections 3.4 and 3.5 describe two important blocks of the control architecture. Several simulations are performed to demonstrate the stability, reliability, flexibility and advantages of our strategy w.r.t. the other methods given in Section 3.5.3. In Section 3.6, our navigation strategy is implemented and experimented in a real urban electric vehicle. Finally, Section 3.7 provides the conclusion of this chapter.

3.1/ PROBLEM STATEMENT

The objective of the navigation task is to lead a UGV towards a specific target in a cluttered environment. This task must be achieved while avoiding static and dynamic obstacles O which can have different shapes. The following scenario is considered (cf. Fig. 3.1):

- Each obstacle O_i in the environment can be surrounded by an elliptical box (cf. Fig. 3.1). The elliptical box is represented by the conic form of an ellipse (cf. Appendix A):
$$\bar{A}\,(x-h)^2 + \bar{B}(x-h)(y-k) + \bar{C}\,(y-k)^2 = 1 \tag{3.1}$$
where $(h,k) \in \mathbb{R}^2$ are the ellipse center coordinates. The $\bar{A}$, $\bar{B}$ and $\bar{C}$ terms are given by:
$$\begin{cases} \bar{A} = & b^{-2}\sin^2(\Omega) + a^{-2}\cos^2(\Omega) \\ \bar{B} = & \left(a^{-2} - b^{-2}\right)\sin(2\Omega) \\ \bar{C} = & b^{-2}\cos^2(\Omega) + a^{-2}\sin^2(\Omega) \end{cases} \tag{3.2}$$
where $a,\ b \in \mathbb{R}^+$ characterize respectively major and minor elliptical semi-axes and Ω gives the ellipse's orientation (cf. Fig. 3.1).
- The robot and target are represented by circles C_R and C_T with radius R_R and R_T respectively (cf. Fig. 3.1).
- D_{RO_i} is the minimal distance between the robot and the obstacle O_i [251].

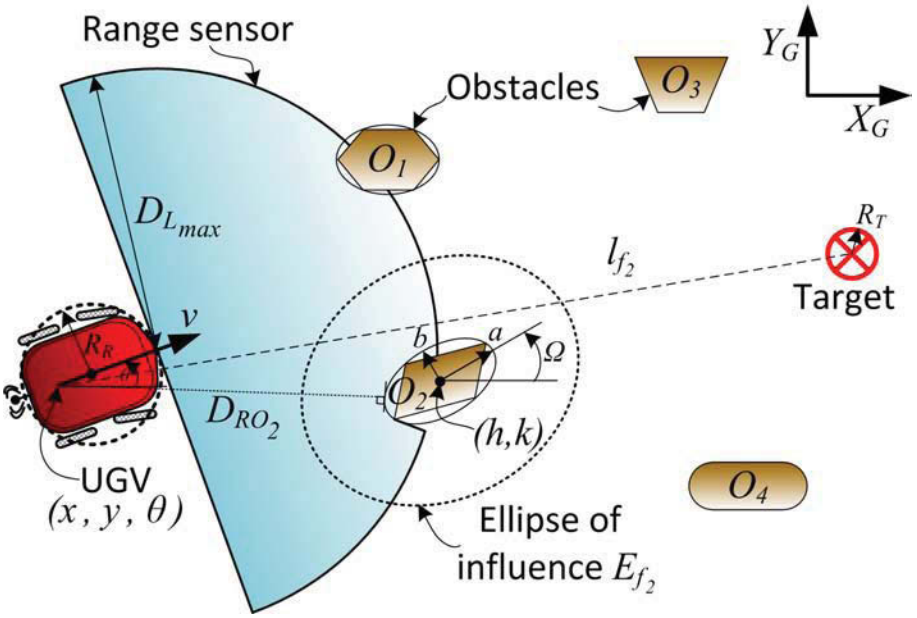


Figure 3.1: The obstacles and robot representation.

- *Ellipse of influence* (E_{f_i}) is an ellipse that has the same center (h, k) and orientation Ω as the ellipse which surround the obstacle O_i (cf. eq. (3.1)). Nevertheless, its major and minor semi-axes, a_{lc} and b_{lc}, are defined as follows:

$$\begin{bmatrix} a_{lc} \\ b_{lc} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} R_R + \delta_E \\ R_R + \delta_E \end{bmatrix} \tag{3.3}$$

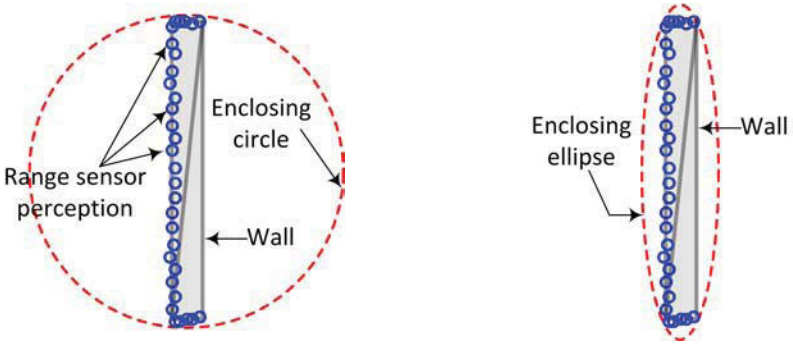
  where δ_E represents safe tolerance margins which include perception uncertainties, feasible control commands and data sensor accuracy. This section focus on reactive navigation, i.e, the ellipse of influence enclosing the detected obstacle is obtained online by the robot.

- l_f as the line passing through the center of C_R and C_T. The intersection points between l_f and E_{f_i} allows to identify if O_i is a dangerous obstacle (which can hinder the robot's future movements) (cf. Fig. 3.1).

The assumption of ellipse boxes instead of circle is to obtain a generic and flexible means to surround accurately different kind of obstacle shapes [293]. One example of shape, which can be properly fit by an ellipse instead of a circle, is a wall (or in general, longitudinal shapes). This example is shown in Fig. 3.2 using perception with uncertainty taken by range sensors on the left side of the wall. Indeed, if this wall is surrounded by a circle, it will have a large radius which can produce more robot's path distance to avoid safely the obstacle [163] (cf. Fig. 3.2(a)). Fig. 3.2(b) shows the ellipse which fits more accurately the dimension of the obstacle.

3.2/ OBSTACLE AVOIDANCE BASED ON LIMIT-CYCLE

In this approach, the robot has to follow accurately limit-cycle trajectories to not collide with obstacles, as detailed in [163, 261]. These authors use a circular limit-cycle characterized by a circle of influence of R_I radius. In [293], it was proposed to extend this



(a) (b)

Figure 3.2: Surrounding a wall using a circle and ellipse shapes.

methodology for more flexible limit-cycle shape (an ellipse). In fact, the ellipse is the generalization of the circle which can be easily obtained by setting $a = b$ in eq. (3.2). The main ideas of this approach are detailed below (cf. Appendix B.1, for more details)

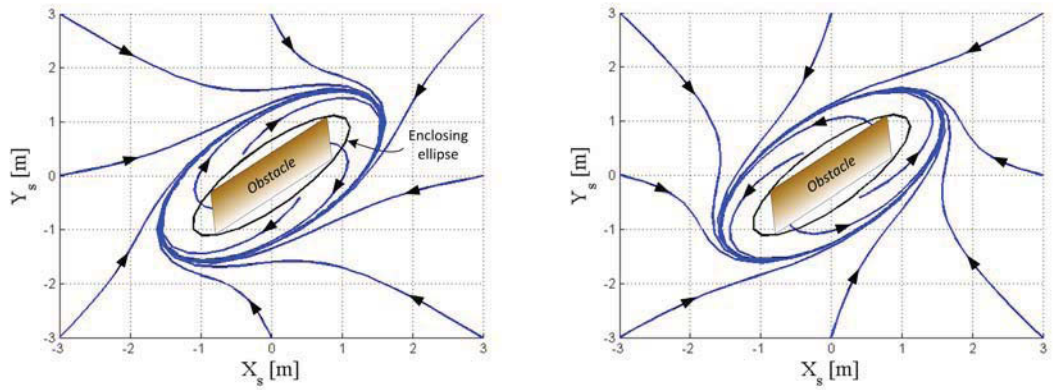
The differential equations of the elliptical limit-cycles are given by:

$$\dot{x}_s = m(\bar{C}_{lc} y_s + 0.5\bar{B}_{lc} x_s) + \mu x_s(1 - \bar{A}_{lc} x_s^2 - \bar{B}_{lc} x_s y_s - \bar{C}_{lc} y_s^2) \tag{3.4}$$

$$\dot{y}_s = -m(\bar{A}_{lc} x_s + 0.5\bar{B}_{lc} y_s) + \mu y_s(1 - \bar{A}_{lc} x_s^2 - \bar{B}_{lc} x_s y_s - \bar{C}_{lc} y_s^2) \tag{3.5}$$

with $m = \pm 1$ according to the avoidance direction (clockwise or counter-clockwise) (cf. Fig. 3.3) and μ is related to the smoothness of the trajectory [293]. (x_s, y_s) corresponds to the robot's position according to the center of the ellipse of influence (h, k) (cf. Fig. 3.1). The variables $\bar{A}_{lc}$, $\bar{B}_{lc}$ and $\bar{C}_{lc}$ are computed using eq. (3.2) with the semi-axes (a_{lc} and b_{lc}) and the orientation Ω of the ellipse of influence E_{f_i}.

The limit-cycle trajectories given by eq. (3.4) and (3.5) converge to the ellipse of influence of the obstacle (demonstration is given in Appendix E.3). The obstacle is avoided while the robot tracks accurately these limit-cycle trajectories. In this work, the robot's position is considered at each sample time in the limit-cycle trajectories. The desired vehicle



(a) $m = 1$ (b) $m = -1$

Figure 3.3: a) Clockwise and b) counter-clockwise shapes for the used elliptical limit-cycles.

orientation θ_d is given by the differential equations of the limit-cycle (cf. eq. (3.4) and (3.5)):

$$\theta_d = \arctan\left(\frac{\dot{y}_s}{\dot{x}_s}\right) \tag{3.6}$$

The obstacle avoidance algorithm based on the limit-cycle approach is described in Appendix B.1 [261, 293]. This algorithm guarantee the avoidance of the obstacle by the robot while performing smooth trajectories.

3.3/ PROPOSED CONTROL ARCHITECTURE FOR NAVIGATION IN CLUTTERED ENVIRONMENTS

This section presents the proposed control architecture for UGV's navigation in a cluttered environment (cf. Fig. 3.4). This control architecture is designed for a UGV modeled as a tricycle robot (cf. Subsection 3.3.1). This architecture aims to manage the interactions among elementary behaviors while guaranteeing the stability of the overall control [260]. It allows to obtain safe and smooth UGV's navigation (cf. Section 3.6).

The UGV's navigation is operated by two elementary behaviors (*Target reaching* and *Obstacle avoidance*); at each sample time one of them is activated by *Hierarchical action selection* (executive layer (cf. Subsection 1.3.3) according to the perceived environment (*Perception* block). Each elementary controller (cf. Fig. 3.4) provides as output an error state $(e_x,\ e_y,\ e_\theta)$ and a desired velocity v_T to the *Control law* block. The proposed control architecture is designed to use a single control law for the overall elementary behaviors.

The proposed control law, designed for a tricycle robot, considers the vehicle poses and velocities. This control law allows the UGV to reach a static or dynamic target with a desired orientation and velocity (cf. Section 3.5). The inputs of the control law (pose errors between the vehicle and target) are provided by the elementary behaviors (cf. subsection 3.3.3 and 3.3.4). The blocks of the proposed architecture are detailed below.

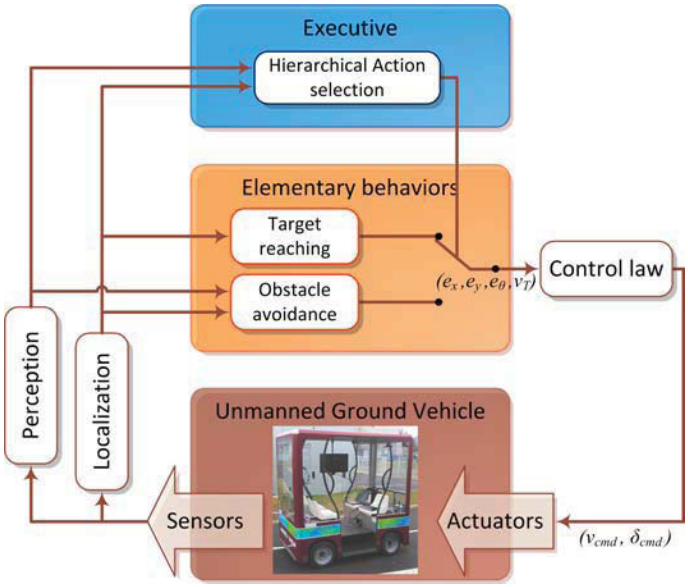


Figure 3.4: Proposed control architecture embedded in the UGV.

This chapter focuses on the development of the *Perception* and *Control law* blocks which are extensively described and analyzed in Section 3.4 and 3.5.

Before describing each block of the control architecture (cf. Fig. 3.4), let us present the used models for the vehicle and target.

3.3.1/ Modeling of Vehicle and Target: Kinematic model

We assume that the UGV evolves in asphalt road and in cluttered urban environment with relatively low speed (less than $3\ m/s$). Hence, the use of kinematic model (which relies on pure rolling without slipping) of the UGV is sufficient. The kinematic model of the UGV is based on the well-known tricycle model [107]. The two front wheels are replaced by a single virtual wheel located at the center between the front wheels. The equations of UGV's model can be written as (cf. Fig. 3.5):

$$\begin{cases} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= v\tan(\gamma)/l_b \end{cases} \tag{3.7}$$

where (x, y, θ) is the UGV's pose in the global reference frame $X_G Y_G$. v and γ are respectively the linear velocity and the orientation of the vehicle front wheel. l_b is the vehicle's wheelbase. I_{cc} is the instantaneous center of curvature of the vehicle's trajectory, $r_c = l_b/\tan(\gamma)$ is the radius of curvature and $c_c = 1/r_c$ is the curvature.

Let us consider a dynamic target modeled as a point (x_T, y_T, θ_T) with non-holonomic constraints (cf. Fig. 3.5). This model allows us to use the general model of robot motion and also to simplify the controller equation. Its kinematic characteristics are given by:

$$\begin{cases} \dot{x}_T &= v_T\cos(\theta_T) \\ \dot{y}_T &= v_T\sin(\theta_T) \\ \dot{\theta}_T &= \omega_T \end{cases} \tag{3.8}$$

where v_T and ω_T are respectively the linear and angular velocities of the target. The radius of curvature is computed by $r_{c_T} = v_T/\omega_T$. An important consideration for target reaching

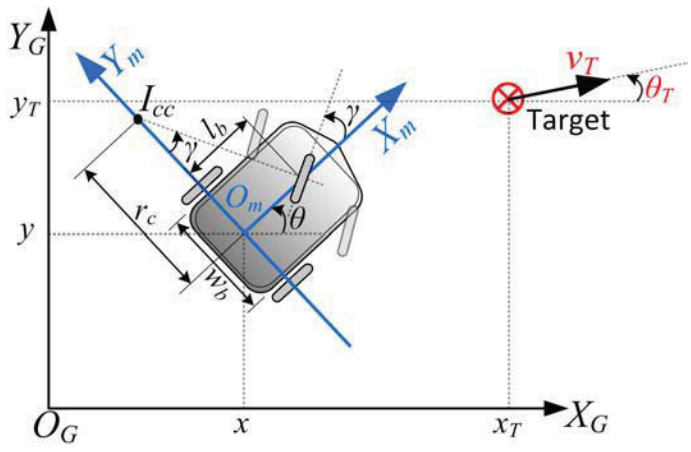


Figure 3.5: Vehicle's and target's configuration in Global $(X_G Y_G)$ and Local $(X_m Y_m)$ reference frames.

is that $v_T \leq v_{max}$ and $r_{c_T} \geq r_{c_{min}}$, where v_{max} and $r_{c_{min}}$ are respectively the maximum linear velocity and the minimum radius of curvature of the vehicle, given by $r_{c_{min}} = l_b / \tan(\gamma_{max})$. For static target reaching (*point stabilization*, i.e., to reach a specific point with a given orientation), ω_T is considered equal to zero and v_T is not necessarily equal to zero; v_T is then considered as a desired velocity value for the vehicle when it reaches the desired target (x_T, y_T, θ_T).

3.3.2/ LOCALIZATION BLOCK

This block uses sensor information to estimate the current pose of the UGV. As mentioned in section 1.3, different methods according to the selected sensors can be used [247, 252, 268]. In this work, the localization is assumed accurate in simulation. For experiments, an RTK-GPS and IMU embedded in each vehicle are combined using EKF to allow accurate estimation of the current configuration (cf. Section 3.6).

3.3.3/ TARGET REACHING BLOCK

This block consists of generating the control variables to drive a UGV (cf. Fig. 3.5 and 3.6) towards a specific target configuration (static or dynamic). Before to introduce the control variables, let us describe the following notations (cf. Fig. 3.6):

- $\left(e_x, e_y, e_\theta\right)$ are the errors w.r.t local frame $X_m Y_m$ between the poses of the UGV and the target.
- d and θ_{RT} are respectively the distance and the angle between the position of the target and the UGV.
- e_{RT} is the error related to the UGV's position (x, y) w.r.t the target's orientation θ_T.

The control variables are linked to the relative pose between the UGV and the target, represented by error state $(e_x,\ e_y,\ e_\theta)$ (cf. Fig. 3.6). The convergence of these variables by the proposed control law allows to guide the UGV towards the target. These errors are

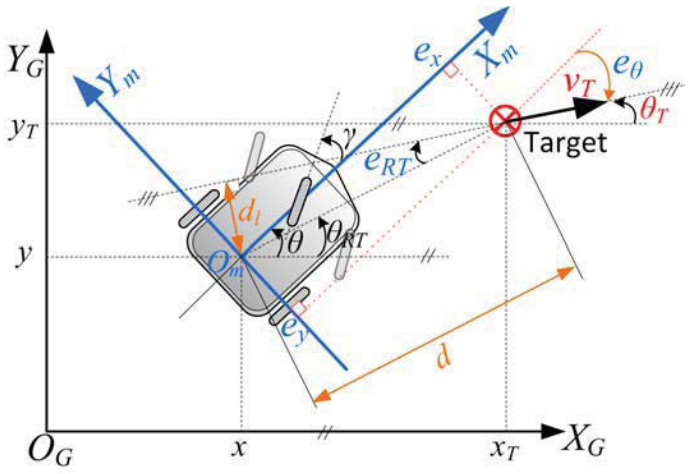


Figure 3.6: Control variables of *Target reaching* block.

computed w.r.t the local reference frame $X_m Y_m$ as follows:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_T - x \\ y_T - y \\ \theta_T - \theta \end{bmatrix} \tag{3.9}$$

The error function e_{RT} is added to the canonical error state eq. (3.9) (cf. Fig. 3.6). Let us first define d and θ_{RT} as (cf. Fig. 3.6):

$$d = \sqrt{(x_T - x)^2 + (y_T - y)^2} \tag{3.10}$$

$$\begin{cases} \theta_{RT} = \arctan\left(\frac{y_T - y}{x_T - x}\right) & \text{if } d > \xi \\ \theta_{RT} = \theta_T & \text{if } d \leq \xi \end{cases} \tag{3.11}$$

where ξ is a small positive value ($\xi \approx 0$).

The error e_{RT} is defined as (cf. Fig. 3.6):

$$e_{RT} = \theta_T - \theta_{RT} \tag{3.12}$$

Furthermore, e_{RT} can be written as a function of e_x, e_y and e_θ as:

$$\begin{aligned} \tan(e_{RT}) &= \tan\left(e_\theta - (\theta_{RT} - \theta)\right) \\ &= \frac{\tan(e_\theta) - e_y e_x^{-1}}{1 + \tan(e_\theta) e_y e_x^{-1}} \\ &= \frac{e_x \tan(e_\theta) - e_y}{e_x + \tan(e_\theta) e_y} \end{aligned} \tag{3.13}$$

where $\tan(\theta_{RT} - \theta) = e_y e_x^{-1}$ (cf. Fig. 3.6). Hence, e_{RT} allows to consider an additional orientation error w.r.t. e_x, e_y and e_θ, e.g., when $e_\theta = 0$ then $e_{RT} = -e_y/e_x$. The stabilization of this error allows to decrease the lateral distance d_l to zero eq. (3.32) (cf. Fig. 3.6), and to always have the UGV in the wake of the target.

Finally, the error state and velocity $(e_x,\ e_y,\ e_\theta,\ v_T)$ are the input of the *Control law* block (cf. subsection 3.5).

3.3.4/ OBSTACLE AVOIDANCE BLOCK

This block generates the control variables to drive a UGV along limit-cycle trajectories to avoid an obstacle (cf. Subsection 3.2). The control variables are errors defined in the previous subsection (e_x, e_y, e_θ) (cf. Fig. 3.6). Their values are given by:

$$e_x = 0 \tag{3.14}$$

$$e_y = 0 \tag{3.15}$$

$$e_\theta = \theta_d - \theta \tag{3.16}$$

where θ_d is the orientation of the limit-cycle trajectory (cf. eq. (3.6)). In this work, $e_x = e_y = 0$ means that UGV's position is at each sample time in the desired position w.r.t. limit-cycle trajectory.

The velocity set-point v_T of limit-cycle trajectory for the control law can be computed as a polynomial function of distance to the obstacle D_{RO}:

$$v_T = f(D_{RO}) \tag{3.17}$$

The coefficients of the polynomial function of degree 2 can be obtained using the UGV's constraints such as the maximum velocity v_{max}, the minimum velocity of the UGV to start and to continue its movement v_{min} and maximum acceleration a_{max}.

3.3.5/ HIERARCHICAL ACTION SELECTION BLOCK

The control architecture uses a *Hierarchical action selection* block to manage the switches between two elementary behaviors, *Target reaching* and *Obstacle avoidance* blocks, according to the environmental perception (cf. Fig. 3.7). The *Hierarchical action selection* block activates the *Obstacle avoidance* block as soon as it detects at least one obstacle which can hinder the future vehicle movement toward its static target (i.e., it exists at least one intersection point between the line l_f and the ellipse of influence E_{f_i} (cf. Fig. 3.1)) [261]. This block allows to anticipate the activation of obstacle avoidance behavior and to decrease the time to reach the assigned target (especially in very cluttered environments), instead of, activate the obstacle avoidance only when the robot is in the immediate vicinity of the obstacle (i.e. $D_{RO} \leq R$ "a certain radius value") [17, 207].

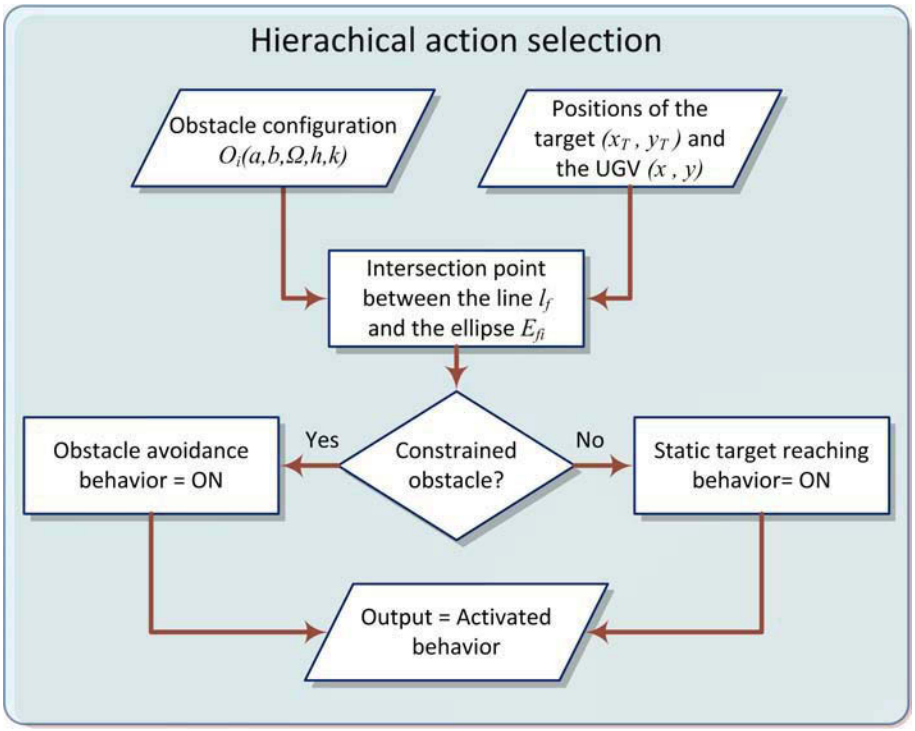


Figure 3.7: Flowchart of the *Hierarchical action selection* block.

3.4/ PERCEPTION BLOCK FOR OBSTACLE DETECTION

The *Perception* block incorporates propriocetive and exteroceptive sensors such as range sensors, cameras, odometers and RTK-GPS. Its goal is to capture information related to the robot's environment, mainly potential obstacles [172, 228, 402] (cf. Fig. 3.8). In [240], the authors propose a method for pedestrian detection using a LIDAR and a camera. The features from LIDAR and vision space are combined in a single vector for a posterior classification.

In this work, the UGV's perception uses data from a range sensor LIDAR (cf. Subsection 3.4.1). The observed data allows to identify online hinder obstacles (cf. Appendix C.2 and C.3) and surround them with the closest ellipse (cf. Subsection 3.4.2) to apply the obstacle avoidance approach based on elliptical limit-cycle [3]. Each process of the *Perception* block is shown in Fig. 3.8. The following sections focus on the used sensor model and the development of *Enclosing data with an ellipse* block. The used functions and methodologies of *Filtering* and *Segmentation* blocks are described in Appendix C.

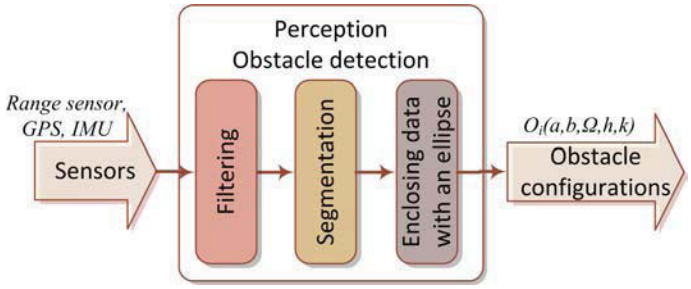


Figure 3.8: Internal block diagram of the *Perception* block.

3.4.1/ RANGE SENSOR MODEL

The sensor specifications and its real performance have significant differences [42, 265]. Mainly, the accuracy of the range sensor, e.g., short range readings are more accurate than the long range ones. In [265], the authors show how the mean and the standard deviation of the errors between the real and the measured range tend to increase with distance.

The model considers a set of n points in $\mathbb{R}^2$ with coordinates $\mathbf{r}_i = (x_i, y_i)^T$ (cf. Fig. 3.9). These points are computed using eq. (C.2) with range data of the UGV given in polar coordinates (D_{r_i}, β_i), where D_{r_i} is the distance between the center of the UGV's range sensor and its impact point and β_i is the orientation with respect to the orientation of the center of UGV's range sensor (cf. Fig 3.9).

The reading data provided by the range sensor at each step-time is modeled by the Normal distribution $D^t_{r_i} = N(\bar{D}^t_{r_i}, P^t_{r_i})$, where $\bar{D}^t_{r_i} = \left(\bar{D}_{r_i}, 0\right)^T$ is the mean vector and $P^t_{r_i} = diag\left(\left[\sigma^2_{D_{r_i}}, \sigma^2_{\beta_i}\right]\right)$ is the covariance defined by the model of the range and angular uncertainties. The range and angular uncertainties are respectively related to the accuracy of the sensor and its opening angle β_i (cf. Fig. 3.9) [265]. The representation of the range data in Cartesian frame is given by:

$$\mathbf{r}_i = \bar{r}_i + \mathbf{v}_i \tag{3.18}$$

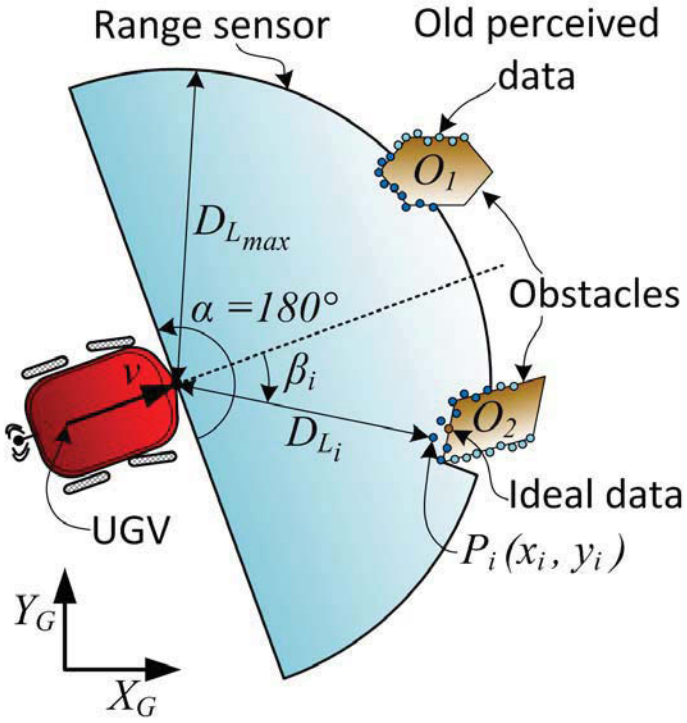


Figure 3.9: Range sensor model and its set of n points.

where $\mathbf{r}_i$ is the point with noise, $\bar{r}_i$ is the point without noise, the noise $\mathbf{v}_i$ is assumed Gaussian with mean $E[\mathbf{v}_i] = 0$ and mean $E[\mathbf{v}_i \mathbf{v}_i^T] = \mathbf{R}_{\mathbf{v}_i}$. The covariance $\mathbf{R}_{\mathbf{v}_i}$ is given by:

$$\mathbf{R}_{\mathbf{v}_i} = \begin{bmatrix} \cos(\beta_i) & -D_{r_i}\sin(\beta_i) \\ \sin(\beta_i) & D_{r_i}\cos(\beta_i) \end{bmatrix} \begin{bmatrix} \sigma^2_{D_{r_i}} & 0 \\ 0 & \sigma^2_{\beta_i} \end{bmatrix} \begin{bmatrix} \cos(\beta_i) & -D_{r_i}\sin(\beta_i) \\ \sin(\beta_i) & D_{r_i}\cos(\beta_i) \end{bmatrix}^T \tag{3.19}$$

3.4.2/ ENCLOSING DATA WITH AN ELLIPSE

The obtained data after processing the range data from LIDAR (elimination of noisy and outliers and segmentation using respectively *Filtering* and *Segmentation* blocks) is enclosed with an ellipse to execute the elliptical limit-cycle approach for reactive obstacle avoidance [3].

Different techniques have been proposed in the literature to enclose data with an ellipse [51, 97, 119]. In [51], the author proposed a technique to obtain the smallest enclosing ellipse for a set of data using the Welzl's algorithm [51] with linear increasing time with regards to data dimension. In [97], the author presents a summary of methods to fit a set of data with an ellipse. The presented methods are the least square fitting based on algebraic and Euclidean distance, Kalman filtering method and robust estimation. In [119], the authors constructed an ellipse using the mean and covariance of the data and the Mahalanobis distance to analyze the relationship among them. The maximum Mahalanobis distance is used for different purposes, namely for outliers detection.

This subsection proposes an online efficient heuristic method based on the Euclidean distance estimation [1, 2, 3] to compute the ellipse parameters (cf. Subsection 3.4.2.1). It is demonstrated that this process ensures that all range data are surrounded by the

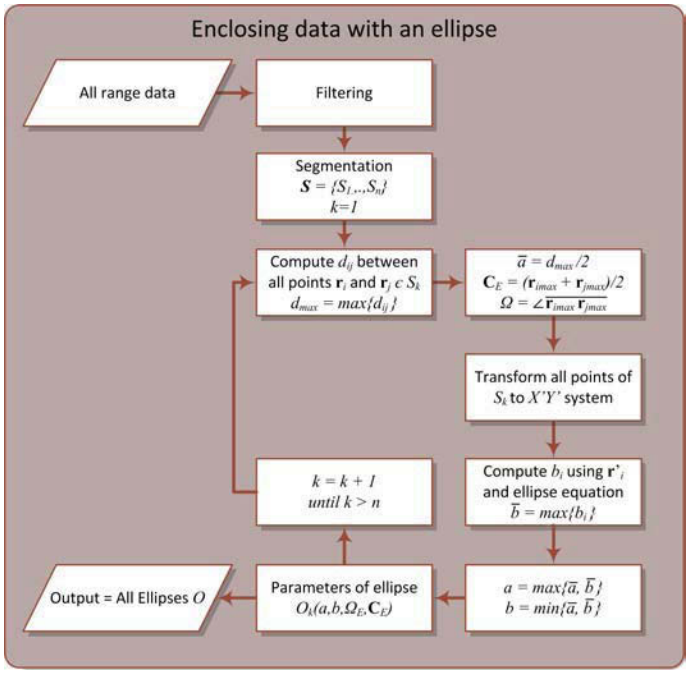


Figure 3.10: Flowchart of the *Enclosing data with an ellipse* block.

computed ellipse. It is important to notice that the above cited literature methods in some cases do not surround all range data (cf. Fig. 3.12). In Subsection 3.4.2.2, a comparative study w.r.t. other techniques based on Least Square and Covariance (cf. Appendix A.5) is presented.

3.4.2.1/ PROPOSED HEURISTIC APPROACH

The Heuristic approach is an efficient method to enclose data with an ellipse. This approach uses the distance between the points to obtain one of the axes and the obtained ellipse encloses all points regardless of the obstacle shape (cf. Fig. 3.10). An important condition of this approach is that it needs to start at least three different points.

Lemma 1. *Consider a set of $n > 2$ points in $\mathbb{R}^2$ with coordinates $\boldsymbol{r}_i(x_i, y_i)$ with $i = 1, \ldots, n$. The parameters of the ellipse that enclose all points are computed as follow (cf. Fig. 3.10):*

- *Compute the distance between all points $d_{ij} = \|r_i - r_j\|$ with $i, j = 1, \ldots, n$; and select the maximum distance d_{max}. This d_{max} is not decreasing if more data points are added.*
- *The ellipse's center $\boldsymbol{C}_E(h,k)$ is the middle point between the points with maximum distance and the orientation of the line that joins these two points define the ellipse's orientation Ω (cf. Fig 3.4.2.1). The first semi-axis is given by $\bar{a} = d_{max}/2$.*
- *Transform n points to the new coordinates system $X'Y'$ (cf. Fig. 3.4.2.1) using eq. (3.20) to obtain the second ellipse semi-axis $\bar{b}$.*

$$\boldsymbol{r}'_i = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) \\ -\sin(\Omega) & \cos(\Omega) \end{bmatrix} (\boldsymbol{r}_i - \boldsymbol{C}_E) \tag{3.20}$$

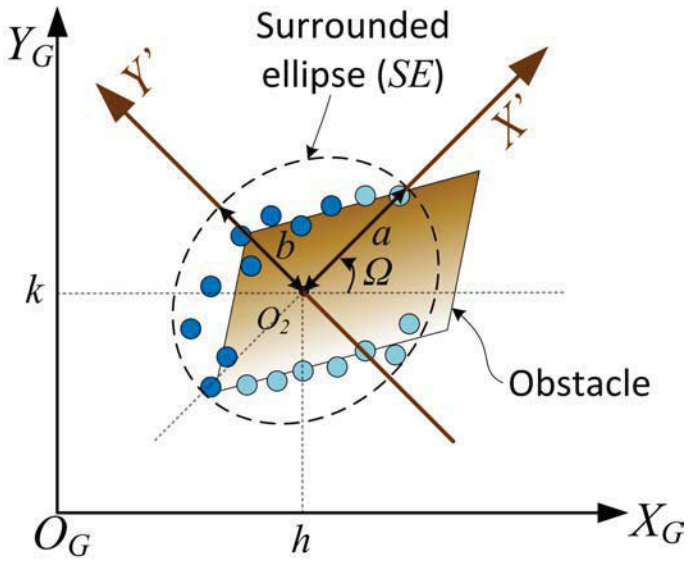


Figure 3.11: Obtained ellipse using the proposed Heuristic approach.

$\mathbf{r}'_i(x'_i, y'_i)$ *are the coordinates in the new reference system.*

- *Compute the semi-axis* b_i *from the standard form of the ellipse (cf. eq. (A.2)) using* $\mathbf{r}'_i$*. The large value of* b_i *is avoided using the points which* $|y'_i|$ *is greater than a threshold* $\epsilon_E > 0$*. This threshold allows to eliminate points that are collinear with first axis* $\bar{a}$ *and points in the perpendicular line to* $\bar{a}$ *at their extremes. The second semi-axis is given by* $\bar{b} = max\{b_i\}$*.*

- *Finally, the semi-axes of the ellipse eq. (A.2), are obtained:*

$$\begin{aligned} a &= max\left\{\bar{a}, \bar{b}\right\} \\ b &= min\left\{\bar{a}, \bar{b}\right\} \end{aligned} \tag{3.21}$$

and, the orientation of the ellipse is $\Omega_E = \Omega + \Pi/2$ *when* $\bar{b}$ *is the major axis, otherwise* $\Omega_E = \Omega$*.*

Proof. *To demonstrate that the proposed Heuristic approach encloses all points, a data point* r_i *which belongs to the set of* n *points is analyzed. This set of* n *points was used to compute parameters of the Ellipse. It is assumed that* $r_i \notin$ *Ellipse then* $d_{ij} = \|r_i - r_j\|$ *(*r_j *belongs also to the set of* n *points) has two cases:*

- $d_{ij} \geq d_{max}$*, however, the first axis of ellipse is the maximum distance between all points. Therefore,* d_{ij} *is the new first axis and* $r_i \in$ *Ellipse.*

- $d_{ij} < d_{max}$*, and the second axis satisfy* $b_i^2 < b^2$*. Using eq. (3.20) and (A.2), it is obtained:*

$$\begin{aligned} b_i^2 < b^2 \Rightarrow \frac{y_i'^2}{1 - \frac{x_i'^2}{a^2}} &< b^2 \\ \frac{x_i'^2}{a^2} + \frac{y_i'^2}{b^2} &< 1 \end{aligned} \tag{3.22}$$

Therefore, r_i *satisfy the ellipse equation (*r_i *is inside* SE*) for* $i = 1, \ldots, n$*.*

It is demonstrated that the point r_i belongs to the Ellipse which was computed using the set of n points. □

3.4.2.2/ COMPARATIVE STUDY

To demonstrate the efficiency of the proposed Heuristic approach to enclose an obstacle with an ellipse, a comparative survey w.r.t. other approaches based on Least Square and Covariance (cf. Appendix A.5) for identification of the closest surrounded ellipse is presented.

This experiment considers the data from a UGV with a frontal range sensor (LIDAR) which has maximum detected range equal to $D_{L_{max}} = 10\ m$ and scanning angle equal to $180°$ (cf. Fig. 3.12). The maximum linear velocity of the UGV is $2.5\ m/s$ and the sample time is $0.01\ s$.

In Fig. 3.12, Heuristic, Least square and covariance approaches for enclosing ellipse of all range data are shown. The UGV was drove around static obstacles. In Fig. 3.12(c), it is noted that the Least square approach has a large obtained ellipse and the proposed Heuristic approach encloses all points with a small area w.r.t. the others.

Fig. 3.12, (b) and (c) show the evolution of the identified ellipse for the Heuristic, Covariance and Least square approaches. We observe that the ellipse's shape of Least square and Covariance approaches change abruptly. Fig. 3.13 shows the identified ellipse's pa-

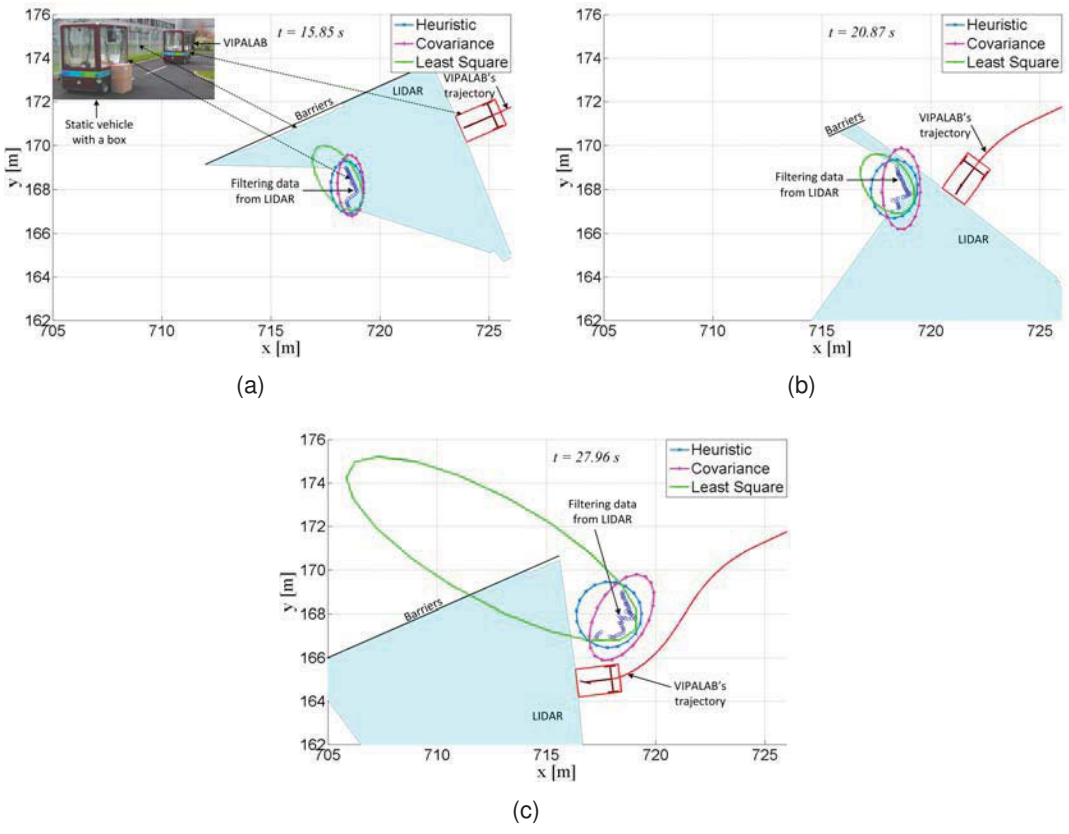


Figure 3.12: Evolution of the ellipse using Heuristic, Covariance and Least Square approaches.

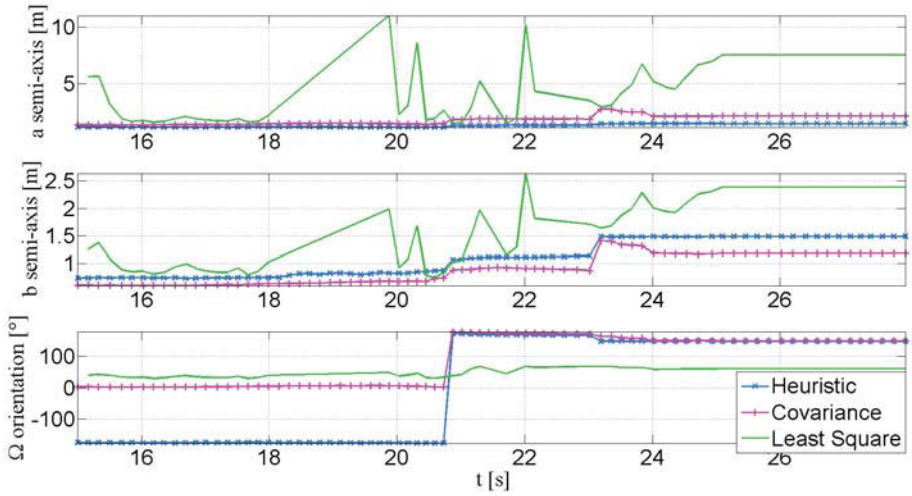


Figure 3.13: Parameters of the estimated ellipses.

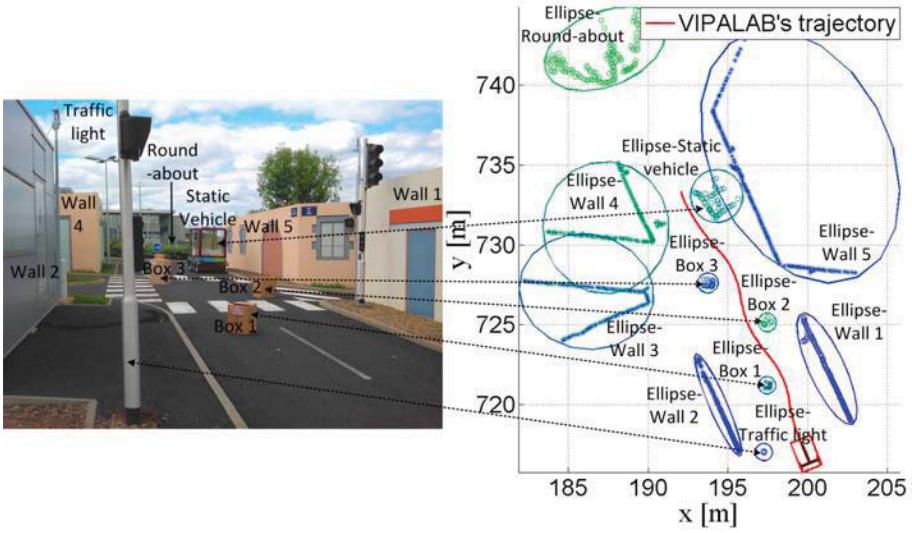


Figure 3.14: Example of the application of the proposed enclosing ellipse method.

rameters using the different approaches. It is noted that the heuristic parameters change smoother w.r.t. others.

Fig. 3.14 shows an example of the application of proposed Heuristic method using filtering data from a LIDAR sensor (cf. Fig. C.6).

3.5/ CONTROL LAW BLOCK

Different control methods for trajectory tracking and path-following dedicated to wheeled mobile robots (unicycle, car-like robot, etc.) have been proposed in the literature [194, 318, 334, 398]. In [37, 318], [3], nonlinear control laws for trajectory tracking are synthesized for a unicycle robot using Lyapunov stability analysis. The Lyapunov functions used in these studies are based only on distance and orientation errors. A trajectory tracking control for a farm vehicle, incorporating sliding in the kinematic model,

is proposed in [194]. For the path-following problem, a control law for a tricycle robot is proposed in [77, 334]. They are based on feedback linearization and chained form representation [132]. The path-following controller allows thus to make the lateral and longitudinal control of the vehicle independent along the reference trajectory. Furthermore, the path-following controller allows smoother convergence to the desired path than the trajectory tracking controller (designed for a time-parametrized trajectory) [244]. The trajectory tracking controller allows to track the trajectory with a desired velocity profile, while the path-following controller acts only on the orientation to drive it along the path. Both, path-following and trajectory tracking controllers require the pose of the closest point to the trajectory (w.r.t. robot configuration) and/or the value of curvature at this point (cf. Fig. 3.17) at each sample time [132, 318]. Although there exist a multitude of techniques to compute these parameters, they can add an error in certain situations thereby influencing negatively the mobile robot control [194, 244].

In this section, the proposed control law based on a novel definition of control variables and Lyapunov function is presented. This Lyapunov function is based on distance and orientation errors and a new parameter related to the angle between the robot and the target position. The synthesized control law can perform either static or dynamic target reaching using only its current pose and velocity. Using dynamic target reaching, trajectory following can also be performed. The control law exhibits good flexibility properties and it could be adapted to diverse autonomous robotic applications such as multi-robot formation (cf. Chapter 5).

The control law generates the desired linear velocity v and front wheels' steering angle γ of the vehicle which lead the errors $(e_x,\ e_y,\ e_\theta)$ (cf. Fig. 3.6) to converge to zero. It is designed as follows:

$$v = v_T \cos(e_\theta) + v_b \tag{3.23}$$

$$\gamma = \arctan(l_b c_c) \tag{3.24}$$

where v_b and c_c are given by:

$$v_b = K_x \left[K_d e_x + K_l d \sin(e_{RT}) \sin(e_\theta) + K_o \sin(e_\theta) c_c\right] \tag{3.25}$$

$$\begin{aligned} c_c = &\frac{1}{r_{c_T} \cos(e_\theta)} + \frac{d^2 K_l \sin(e_{RT}) \cos(e_{RT})}{r_{c_T} K_o \sin(e_\theta) \cos(e_\theta)} + K_\theta \tan(e_\theta) \\ &+ \frac{K_d e_y - K_l d \sin(e_{RT}) \cos(e_\theta)}{K_o \cos(e_\theta)} + \frac{K_{RT} \sin^2(e_{RT})}{\sin(e_\theta) \cos(e_\theta)} \end{aligned} \tag{3.26}$$

$\mathbf{K} = (K_d, K_l, K_o, K_x, K_{RT}, K_\theta)$ is a vector of positive constants which must be defined by the designer (cf. Subsection 3.5.2). K_d, K_l and K_o are respectively related to the desired convergence of distance, lateral and angular errors w.r.t. the assigned target. K_x, K_{RT} and k_θ are related to the maximum linear and angular velocities [4, 8].

The stability of the proposed control law is proved in next subsection. An error boundary, enabling maximum error estimation according to the definition of the controller parameters, is computed in subsection 3.5.2. With this information, vehicle navigation is safe within a certain boundary. A brief description and comparison of some navigation methods from the literature are given in subsection 3.5.3.1.

3.5.1/ STABILITY OF THE CONTROL LAW

In this subsection, the stability of the derivatives of the errors $(e_x,\ e_y,\ e_\theta)$ (3.9 on page 54) and e_{RT} (3.12 on page 54), which are computed using eq. (3.7), (3.8), (3.10) and (3.11), is analyzed. The differential error system is obtained by:

$$\begin{aligned}
\dot{e}_x &= \cos(\theta)(\dot{x}_T - \dot{x}) + \sin(\theta)(\dot{y}_T - \dot{y}) - \sin(\theta)(x_T - x)\dot{\theta} + \cos(\theta)(y_T - y)\dot{\theta} \\
&= -v + e_y\dot{\theta} + v_T[\cos(\theta_T)\cos(\theta) + \sin(\theta_T)\sin(\theta)] \\
&= -v + e_y v\tan(\gamma)l_b^{-1} + v_T\cos(e_\theta) \qquad (3.27) \\
\dot{e}_y &= -\sin(\theta)(\dot{x}_T - \dot{x}) + \cos(\theta)(\dot{y}_T - \dot{y}) - \cos(\theta)(x_T - x)\dot{\theta} - \sin(\theta)(y_T - y)\dot{\theta} \\
&= -e_x\dot{\theta} - v_T\cos(\theta_T)\sin(\theta) + v_T\sin(\theta_T)\cos(\theta) \\
&= -e_x v\tan(\gamma)l_b^{-1} + v_T\sin(e_\theta) \qquad (3.28)
\end{aligned}$$

$$\begin{aligned}
\dot{e}_\theta &= \dot{\theta}_T - \dot{\theta} \\
&= \omega_T - \omega \\
&= \frac{v_T}{r_{c_T}} - v\tan(\gamma)l_b^{-1} \qquad (3.29) \\
\dot{e}_{RT} &= \dot{\theta}_T - \dot{\theta}_{RT} \\
&= \frac{v_T}{r_{c_T}} - \frac{d}{dt}\left[\arctan\left(\frac{y_T - y}{x_T - x}\right)\right] \\
&= \frac{v_T}{r_{c_T}} - v_T\frac{\sin(\theta_T)(x_T - x) - \cos(\theta_T)(y_T - y)}{d^2} \\
&\quad - \frac{-v\sin(\theta)(x_T - x) + v\cos(\theta)(y_T - y)}{d^2} \\
&= \frac{v_T}{r_{c_T}} - \frac{v_T e_x\sin(e_\theta)}{d^2} + \frac{v_T e_y\cos(e_\theta)}{d^2} - \frac{e_y v}{d^2} \qquad (3.30)
\end{aligned}$$

Assumption 1. *The subsequent development is based on the assumption that the initial values of e_{RT} and e_θ satisfy:*

$$e_{RT} \in]-\pi/2, \pi/2[\quad and \quad e_\theta \in]-\pi/2, \pi/2[\qquad (3.31)$$

These conditions guarantee that the target is ahead to the vehicle with respect to its orientation.

Theorem 1. *The control law given by eq. (3.23) and (3.24) ensures that the differential system (cf. eq. (3.27), (3.28), (3.29) and (3.30) is asymptotically stable according to Lyapunov-based analysis if the Assumption 1 is satisfied [147].*

Proof. *The stability of the system is analyzed using Lyapunov method [147]. The proposed Lyapunov function V, given by eq. (3.32), is a function of three parameters which depend on the distance d between the target and vehicle positions, the distance d_l from the vehicle to the target line (line which passes through the target position with an orientation equal to the target orientation; this term is related to the Line of Flight and Sight of the target [183]), and the orientation error e_θ between the vehicle and the target (cf. Fig. 3.6).*

The candidate Lyapunov function V is a positive-definite function [147] when considering eq. (3.31) (Assumption 1). The Lyapunov function is given by:

$$\begin{aligned} V =& \frac{1}{2}K_d d^2 + \frac{1}{2}K_l d_l^2 + K_o[1-\cos(e_\theta)] \\ =& \frac{1}{2}K_d d^2 + \frac{1}{2}K_l d^2 \sin^2(e_{RT}) + K_o[1-\cos(e_\theta)] \end{aligned} \tag{3.32}$$

This Lyapunov function can be written with respect to e_x, e_y as follows:

$$V = \frac{1}{2}\left(e_x^2 + e_y^2\right)[K_d + K_l \sin^2(e_{RT})] + K_o[1-\cos(e_\theta)] \tag{3.33}$$

To guarantee the stability of the system, $\dot{V}$ must be negative-definite [147]. By taking the derivative of eq. (3.33) and using eq. (3.27), (3.28), (3.29), (3.30), (3.23) and (3.24), $\dot{V}$ can be written:

$$\begin{aligned} \dot{V} =& (e_x\dot{e}_x + e_y\dot{e}_y)[K_d + K_l \sin^2(e_{RT})] + K_l d^2 \sin(e_{RT})\cos(e_{RT})\dot{e}_{RT} + K_o \sin(e_\theta)\dot{e}_\theta \\ =& (e_x\left[e_y v c_c - v + v_T\cos(e_\theta)\right] + e_y\left[v_T\sin(e_\theta) - e_x v c_c\right])[K_d + K_l\sin^2(e_{RT})] \\ &+ K_l d^2 \sin(e_{RT})\cos(e_{RT})\left[\frac{v_T}{r_{c_T}} - \frac{v_T e_x \sin(e_\theta)}{d^2} + \frac{v_T e_y\cos(e_\theta)}{d^2} - \frac{e_y v}{d^2}\right] \\ &+ K_o\sin(e_\theta)\left(\frac{v_T}{r_{c_T}} - v c_c\right) \end{aligned} \tag{3.34}$$

Substituting eq. (3.23) in (3.34)

$$\begin{aligned} \dot{V} =& \left[-e_x v_b + v_T e_y \sin(e_\theta)\right][K_d + K_l\sin^2(e_{RT})] + K_o\sin(e_\theta)\left[\frac{v_T}{r_{c_T}} - v_T\cos(e_\theta)c_c - v_b c_c\right] \\ &+ K_l\sin(e_{RT})\cos(e_{RT})\left[d^2\frac{v_T}{r_{c_T}} - v_T e_x\sin(e_\theta) - e_y v_b\right] \\ =& \left[e_y(K_d + K_l\sin^2(e_{RT})) - e_x K_l\sin(e_{RT})\cos(e_{RT})\right]v_T\sin(e_\theta) \\ &+ \frac{v_T}{r_{c_T}}\left[d^2 K_l\sin(e_{RT})\cos(e_{RT}) + K_o\sin(e_\theta)\right] - v_T K_o\sin(e_\theta)\cos(e_\theta)c_c \\ &- v_b\left[e_x(K_d + K_l\sin^2(e_{RT}))\right] - v_b\left[e_y K_l\sin(e_{RT})\cos(e_{RT}) + K_o\sin(e_\theta)c_c\right] \end{aligned} \tag{3.35}$$

Using eq. (3.13) in the first and last terms of eq. (3.35) and factorizing the common terms, it holds that:

$$\begin{aligned} \dot{V} =& v_T\sin(e_\theta)[K_d e_y - K_l d\sin(e_{RT})\cos(e_\theta)] + \frac{v_T}{r_{c_T}}[d^2 K_l\sin(e_{RT})\cos(e_{RT}) + K_o\sin(e_\theta)] \\ &- v_b[K_d e_x + K_l d\sin(e_{RT})\sin(e_\theta) + K_o\sin(e_\theta)c_c] - v_T K_o\sin(e_\theta)\cos(e_\theta)c_c \end{aligned} \tag{3.36}$$

Finally, using eq. (3.25) and (3.26) in (3.36), it is obtained:

$$\begin{aligned} \dot{V} =& -K_x[K_d e_x + K_l d\sin(e_{RT})\sin(e_\theta) + K_o\sin(e_\theta)c_c]^2 \\ &- v_T K_o K_\theta\sin^2(e_\theta) - v_T K_o K_{RT}\sin^2(e_{RT}) \leq 0 \end{aligned} \tag{3.37}$$

Equation (3.37) shows that the system is stable while the initial conditions eq. (3.31) are satisfied. To ensure the asymptotic stability of the proposed control law, $\dot{V}$ has to be a

negative-definite function. Let us consider two cases, one where $\dot{V} = 0$ *with* $v_T > \xi$ *and another with* $v_T = \xi$*, where* ξ *a constant value (*$\xi \approx 0$*). Firstly, when* $v_T > \xi$ *and using the initial assumption* $\mathbf{K} > 0$*, it is straightforward to show that* $e_x,\ e_\theta,\ e_{RT}$ *are equal to zero to satisfy eq. (3.37); then according to eq. (3.11), (3.12) and (3.31),* d *is equal to zero (*$e_x = e_y = 0$*). Hence,* $\dot{V}$ *is equal to zero when* $v_T > \xi$*, only if* $(e_x, e_y, e_\theta) = (0, 0, 0)$*.*

Secondly, let us consider the case where $v_T = \xi$*. The initial assumption is identical. Therefore, the second and third terms of eq. (3.37) are equal to zero when* $v_T = \xi$*. Additionally, when* $v_T = \xi$ *(quasi-static case) then it is considered* $r_{c_T} \to \infty$ *(cf. Subsection 3.3.1); consequently the first term of* $\dot{V}$ *is equal to zero when:*

$$K_d e_x + K_l d \sin(e_{RT}) \sin(e_\theta) + K_o \sin(e_\theta) c_c = 0 \tag{3.38}$$

Replacing eq. (3.26) with $r_{c_T} \to \infty$ *in (3.38), the following expression is obtained:*

$$\begin{aligned} 0 =& K_d e_x + K_l d \sin(e_{RT}) \sin(e_\theta) + \tan(e_\theta)[K_d e_y - K_l d \sin(e_{RT}) \cos(e_\theta)] \\ &+ K_o \sin(e_\theta) \left[K_\theta \tan(e_\theta) + \frac{K_{RT} \sin^2(e_{RT})}{\sin(e_\theta) \cos(e_\theta)} \right] \\ =& K_d [e_x + e_y \tan(e_\theta)] + K_o K_\theta \frac{\sin^2(e_\theta)}{\cos(e_\theta)} + K_o K_{RT} \frac{\sin^2(e_{RT})}{\cos(e_\theta)} \end{aligned} \tag{3.39}$$

Using eq. (3.13) in (3.39), we finally obtain:

$$K_d d \frac{\cos(e_{RT})}{\cos(e_\theta)} + K_o K_\theta \frac{\sin^2(e_\theta)}{\cos(e_\theta)} + K_o K_{RT} \frac{\sin^2(e_{RT})}{\cos(e_\theta)} = 0 \tag{3.40}$$

Equation (3.40) exhibits quadratic terms. Consequently, considering the initial conditions eq. (3.31), $\cos(e_{RT})$ *and* $\cos(e_\theta)$ *are greater than zero. Therefore, all the terms of eq. (3.40) are positive and they must be equal to zero (i.e.,* $d,\ e_\theta,\ e_{RT} = 0$*, and if* $d = 0$ *then* $e_x,\ e_y = 0$*). Hence, from eq. (3.40),* $\dot{V}$ *is equal to zero when* $v_T = \xi$ *and* $r_{c_T} \to \infty$*, only if* $(e_x, e_y, e_\theta) = (0, 0, 0)$*.*

In conclusion, if $v_T > \xi$ ***or*** $v_T = \xi$***,*** V ***is always strictly positive and*** $\dot{V}$ ***is always strictly negative while*** $(e_x, e_y, e_\theta) \neq (0, 0, 0)$***. Therefore, the system is asymptotically stable while the initial vehicle conditions eq. (3.31) are satisfied.*** □

3.5.2/ SAFE TARGET REACHING

The synthesis of the proposed control law using a Lyapunov function enables to confirm its asymptotic stability (cf. Subsection 3.5.1). Nonetheless, it does not allow to obtain immediately the error values when the robot is in the immediate vicinity of the target to reach. These errors can be used to select an suitable next target to be accurately reached by the UGV and so on, i.e., a safe navigation through sequential targets placed in suitable position is performed (cf. Section 4.2).

The aim of this subsection is to determine a relation between the upper bound of the errors d and e_θ, denoted respectively E_{dis} and $E_\angle$ (cf. Fig. 3.6 and 3.15) and the controller parameters **K**. These errors will be used as switch conditions between targets for the proposed navigation strategy in structured environments (cf. Subsection 4.2). The vehicle's dynamic and the localization of the vehicle and target are assumed to be always accurate (cf. Subsection 3.3.2).

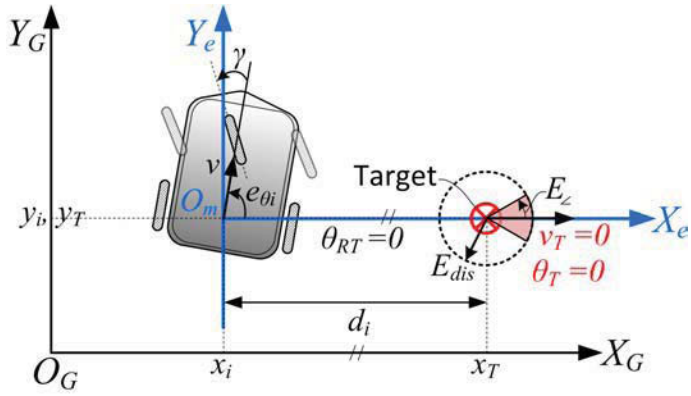


Figure 3.15: Limit vehicle configuration for tuning controller parameters.

The proposed analysis consists of determining the minimum d_i (cf. Fig. 3.15) which allows to satisfy at the same time, the vehicle's kinematic model and the errors ($d(t_f) \leq E_{dis}$ and $e_\theta(t_f) \leq E_{angle}$) when the vehicle reaches the the assigned target (at final time t_f).

Fig. 3.16 shows the global flowchart to obtain the minimal initial distance d_i while having:

- $|e_{\theta_i}|$ and $|e_{RT_i}| < \pi/2$ respecting the stability conditions of the proposed control law (cf. Subsection 3.5.1).
- The constant vector gain $\mathbf{K} = (K_d, K_l, K_o, K_x, K_\theta, K_{RT})$ which characterizes the control law parameters, known by advance. $\mathbf{K}$ is fixed while taking into account vehicle constraints such as maximum vehicle's velocity v_{max} and minimum curvature radius $r_{c_{min}}$.
- The upper bound of the errors (E_{dis} and E_{angle}) to satisfy at $t = t_f$.

To simplify controller analysis, the orientation e_θ and distance d errors are dealt separately. Firstly, the orientation error is computed considering enough initial distance vehicle-target d_i ($d_i \gg E_{dis}$) to permit monotonous convergence of e_θ towards zero. This consideration allows to estimate the minimum time t_f to attain effectively $e_\theta \leq E_{angle}$ (cf. Fig. 3.15). The subsequent analysis considers a static target ($\dot{x}_T = \dot{y}_T = 0$ and $r_{c_T} \to \infty$) and an extreme vehicle's configuration, $|e_{\theta_i}| = \pi/2 - \zeta$ (where ζ is a positive value ≈ 0), hence the vehicle has initially the maximum admissible orientation error with respect to the target (cf. Fig. 3.15 and 3.21). The idea is to use the analysis of this initial limit vehicle's orientation error e_{θ_i}, which correspond to the slowest error convergence, to extrapolate thereafter the result for less critical vehicle's configuration $|e_{\theta_0}| \ll \pi/2$. Indeed, for less critical configuration, the convergence of e_θ will be faster than the limit defined case ($|e_{\theta_i}| \longrightarrow \pi/2$) (cf. Fig. 3.22).

Using the described consideration of Fig. 3.15 and eq. (3.23 on page 62), (3.24 on page 62) and (3.25 on page 62) in (3.29 on page 63), $\dot{e}_\theta$ can be written as:

$$\begin{aligned}
\dot{e}_\theta = \frac{v_T}{r_{c_T}} - v\frac{\tan(\gamma)}{l_b} &= -\left[\frac{K_x K_d d}{\cos(e_\theta)} + K_x K_o K_\theta \frac{\sin^2(e_\theta)}{\cos(e_\theta)}\right]\left[\frac{K_d e_y}{K_o \cos(e_\theta)} + K_\theta \tan(e_\theta)\right] \\
&= -\frac{K_x}{K_o \cos^2(e_\theta)}\left[K_d d + K_o K_\theta \sin^2(e_\theta)\right]\left[K_d d \sin(e_\theta) + K_o K_\theta \sin(e_\theta)\right] \\
&= -\frac{K_x(K_d d + K_o K_\theta)}{K_o \cos^2(e_\theta)}\left[K_d d + K_o K_\theta \sin^2(e_\theta)\right]\sin(e_\theta) \qquad (3.41)
\end{aligned}$$

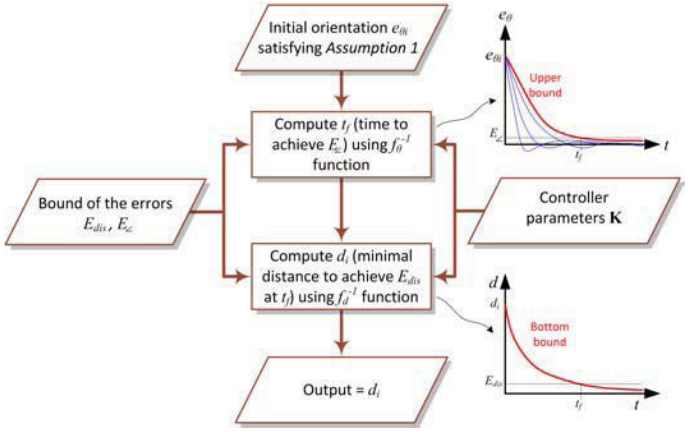


Figure 3.16: Flowchart of the analysis to obtain d_i.

To solve the differential equation (3.41), let us introduce the following notations:

$$\begin{cases} c_1 = & K_d d \\ c_2 = & K_o K_\theta \\ c_3 = & \sqrt{(c_1 c_2^{-1} + 1)} \end{cases} \tag{3.42}$$

It is observed that the value of c_1 depends on the distance d. Since, this analysis focuses in the close proximity of the UGV to the target, the value of d is smaller w.r.t. the initial distance d_i. Then, K_d can be chosen as a function of the initial distance as $K_d = 1/d_i$, which allows to approximate the $A \ll 1$ as a constant value. Hence, the analytic solution of eq. (3.41) (cf. Appendix D), while considering c_1, c_2 and c_3 as constant values, has the following form:

$$\ln\left[\tan\left(\frac{e_\theta}{2}\right)\left(\frac{c_3 + \cos(e_\theta)}{c_3 - \cos(e_\theta)}\right)^{c_3/2}\right]\Bigg|_{e_{\theta_i}}^{e_\theta} = -\frac{K_x c_1 c_2}{K_o} c_3^2 t\Big|_0^t \tag{3.43}$$

From eq. (3.43), the upper bound of e_θ can be written as:

$$e_\theta = f_\theta(t, \mathbf{K}, e_{\theta_i}) \tag{3.44}$$

where:

$$f_\theta = 2\tan\left(\frac{e_{\theta_i}}{2}\right)\left[\frac{(c_3 + \cos(e_{\theta_i}))(c_3 - 1)}{(c_3 - \cos(e_{\theta_i}))(c_3 + 1)}\right]^{c_3/2} e^{-\frac{K_x c_1 c_2}{K_o} c_3^2 t} \tag{3.45}$$

This function approximates enough faithfully the evolution of the error e_θ when the vehicle is close to the target ($d \ll di$) at final time t_f. From eq. (3.44), t_f is obtained:

$$t_f = f_\theta^{-1}(E_\angle, \mathbf{K}, e_{\theta_i}) \tag{3.46}$$

Using equation 3.44 and a fixed value of **K**, it is immediate to compute the time t_f necessary to obtain $e_\theta = E_{angle}$. If $t > t_f$ then e_θ will be certainly $\leq E_{angle}$ (cf. Fig. 3.21).

Secondly, let us use the fixed time t_f to determine d_i which permit to guarantee always $d(t_f) \leq E_{dis}$ and $e_\theta(t_f) \leq E_{angle}$ at final time t_f for any initial vehicle configuration respecting: e_{θ_i}, $e_{RT_i} < \pi/2$. For a fixed navigation time $t = t_f$, the maximal possible initial distance d_i permitting to reach the target, is given when the vehicle's initial configuration corresponds

to: $e_{RT} = 0$ and $e_\theta = 0$ (straight line to the target). Obviously, a greater value of $d(t = 0) > d_i$ increases the certainty that the vehicle (in extreme configuration, as depicted in Fig. 3.15), could reach the target with appropriate $e_\theta \leq E_{angle}$ (cf. Fig. 3.21).

Hence, taking $e_{RT} = 0$ and $e_\theta = 0$, eq. (3.25) can be written as

$$v_b = K_x K_d d \tag{3.47}$$

Introducing eq. (3.27), (3.28), (3.23) and (3.47) in the derivative of the distance, we obtain:

$$\begin{aligned} \dot{d} =& \frac{e_x \dot{e}_x + e_y \dot{e}_y}{d} \\ =& -\frac{e_x v_b}{d} \\ =& -K_x K_d d \end{aligned} \tag{3.48}$$

Integrating eq. 3.48, we obtain:

$$\begin{aligned} \int_{d_i}^{d} 1/d \; \partial d = \int_0^t -K_x K_d \; \partial t \\ d = f_d(t, \mathbf{K}, d_i) \end{aligned} \tag{3.49}$$

where:

$$f_d(t, \mathbf{K}, d_i) = d_i e^{-K_x K_d t} \tag{3.50}$$

From eq. (3.49) and (3.50), the convergence of the distance depends on K_x, K_d and the obtained time t_f. Therefore, using equation 3.49 and while knowing that the objective here is to obtain d_i (minimal distance covered by the UGV) which satisfy $d = E_{dis}$ at $t = t_f$, the d_i could be easily computed by:

$$d_i = f_d^{-1}(t_f, \mathbf{K}, E_{dis}) = E_{dis} e^{K_x K_d t_f} \tag{3.51}$$

Moreover, (3.44) and (3.51) show the relations between initial configuration, controller parameters and error of reaching the target. Therefore, for certain initial configuration and controllers parameters, the defined bounded of the errors can be obtained. This analysis is also conclusive for the dynamic target case, where the variation of the distance d between the UGV and target is slower than in the static target case. In the dynamic target case, the control law with the designed parameters $\mathbf{K}$ will have thus more time to converge the errors eq. (3.9) to zero.

3.5.3/ Simulation analysis

The following simulations show the advantages, stability and reliability of the proposed control law to drive a UGV towards a specific target.

3.5.3.1/ Comparative study of the control law

In this subsection, two common approaches to follow a dynamic target by a tricycle are presented and compared with the proposed control law. The comparison is focused on

the position and orientation errors and the convergence time. In the sequel, the dynamic target is assumed to be another identical UGV. These two strategies are briefly described below:

1. **Approach based on a target model** (Chained system): In [132] and [244], a control law to track a reference vehicle (target) is proposed. A variable transformation to obtain the control is applied to the error system and commands. The control law is synthesized using a suitable Lyapunov function (more details can be found in [132]). The desired steering angle is computed by integration. Nevertheless, the control law considers a non-zero linear target velocity, i.e, if the target is static then the commands sent to the vehicle are zero.

2. **Approach based on a reference path** (Frenet control): In [89], a method for following a vehicle based on a Frenet frame was developed (cf. Fig. 3.17). It exploits the use of chained systems to separate the lateral and longitudinal control. Therefore, each controller can be designed independently. The lateral control is obtained using chained transformation (more details are given in [77] and [132]). The longitudinal control consists in keeping a specific curvilinear distance d_s between the target and the UGV. One drawback of this approach is the dependency on a known reference path for the UGV, i.e., if the vehicle follows a dynamic target then the target trajectory must be accurately known by the UGV.

The approaches presented above are implemented in simulation and compared to our proposal. In order to do this, the main target (equivalent to a Leader robot in formation control [4, 5]) tracks a sinusoidal trajectory and the followers must maintain a distance of $5\ m$ w.r.t. this first robot, i.e., the secondary target to be reached is located at $5\ m$ (curvilinear distance) from the main target. Moreover, the different controllers were tuned using the best performance for the task to achieve.

Fig. 3.18 and 3.19 show the trajectories and the control output of each vehicle (leader and followers). It can be noticed that the proposed control law has a similar performance to the controller based on a reference trajectory (Frenet control).

Table 3.1 shows the convergence time to satisfy the error threshold in distance d_{Target} and orientation $e_{\theta_{Target}}$ with regards to the target pose (cf. Fig. 3.6). The proposed control

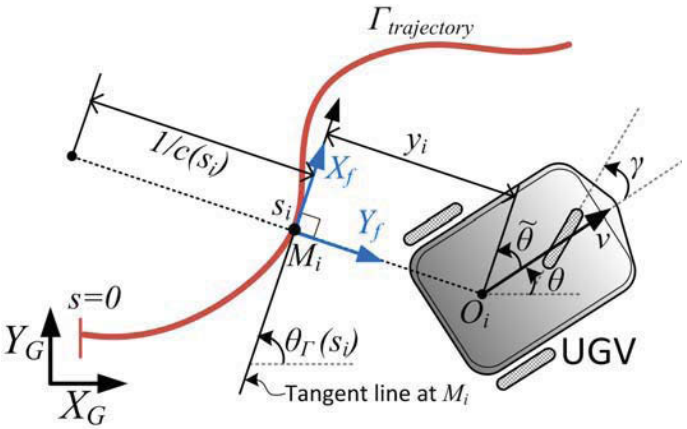


Figure 3.17: Vehicle modeling in Frenet frame.

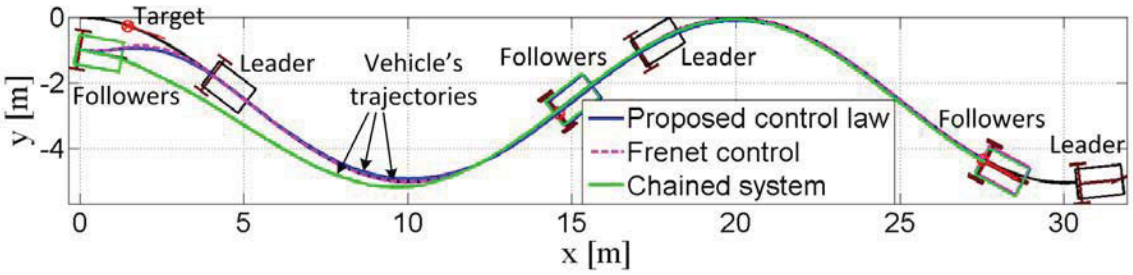


Figure 3.18: Trajectory of the leader and followers (Frenet control [77]) and (Chained system [132]).

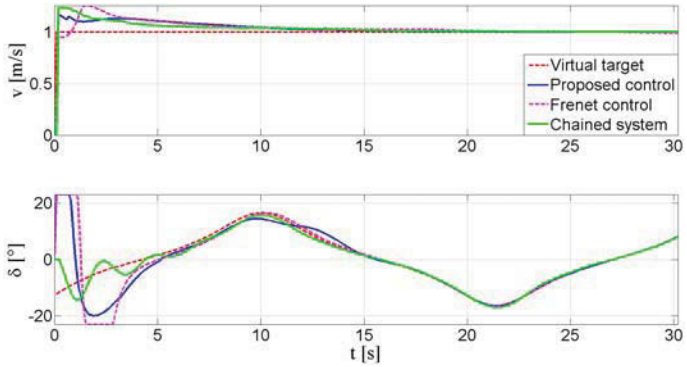


Figure 3.19: Control output.

law has the smallest convergence time to satisfy simultaneously both threshold errors (maximum value between convergence time of d_{Target} and $e_{\theta_{Target}}$). The difference with the Frenet control is equal to $2.53\ s$ and with chained system is equal $2.28\ s$.

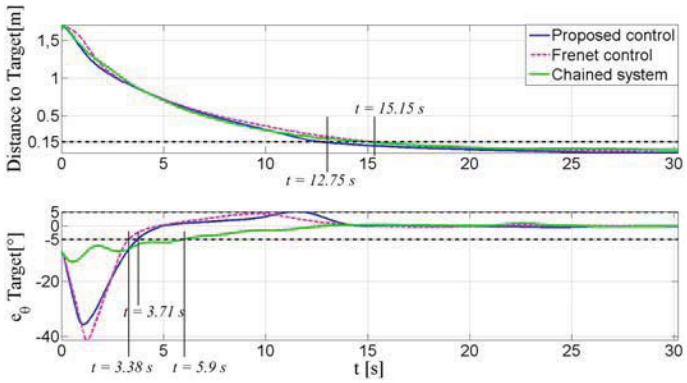
Table 3.1: Comparison with dynamic target position (cf. Fig. 3.6)

Time $[s]$ to reach:	$d_{Target} < 0.15\ m$	$\lvert e_{\theta_{Target}} \rvert < 5°$
Proposed control	$12.75\ s$	$3.71\ s$
Frenet control	$15.28\ s$	$3.28\ s$
Chained system	$15.03\ s$	$5.9\ s$

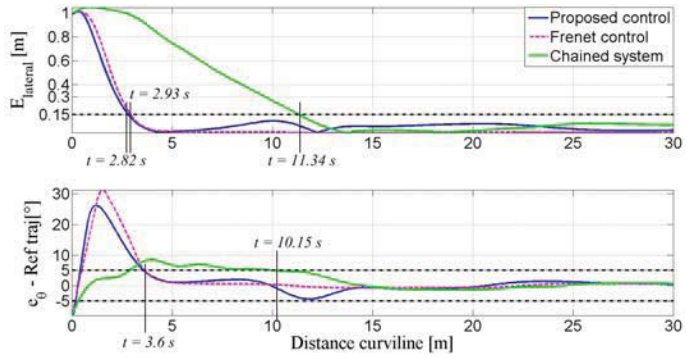
Table 3.2 shows the convergence time to satisfy the error threshold in distance y_i and orientation $\tilde{\theta}$ with regards to the target trajectory (Frenet reference frame $X_f Y_f$ (cf. Fig. 3.17)). The Proposed control law has the smallest convergence time to satisfy simultaneously both threshold errors. There is only a small difference $(0.03\ s)$ compared to Frenet control while the proposed control law uses only the current pose of the target (thus more flexible).

Table 3.2: Comparison of the errors defined according to Frenet frame (cf. Fig. 3.17)

Time $[s]$ to reach:	$\lvert y_i \rvert < 0.15\ m$	$\lvert \tilde{\theta} \rvert < 5°$
Proposed control	$2.82\ s$	$3.58\ s$
Frenet control	$2.93\ s$	$3.61\ s$
Chained system	$11.34\ s$	$10.15\ s$



(a) Errors defined according to Target's pose.



(b) Errors defined according to Frenet reference frame.

Figure 3.20: Simulation results of the comparative study between control laws.

The proposed control law was not designed to take into account the reference trajectory, however the obtained results are very close to those designed for trajectory tracking. In addition to its accuracy, the proposed control law has more flexibility (cf. Section 3.5) to perform the autonomous navigation of the UGVs. Indeed, it is only necessary to know the current pose and dynamics of the target instead of all recorded trajectory.

3.5.3.2/ VALIDATION OF SAFE TARGET REACHING

The following simulations verify the presented analysis in Subsection 3.5.2 and show the performance (safety, smoothness and convergence) of the control law to reach a desired final configuration (pose and velocity). The UGV starts at the same position with different initial distance and orientations.

The first simulation validates the analysis presented in subsection 3.5.2, where the minimum value of d_i, obtained with a limit vehicle configuration $e_\theta \approx \pi/2$, allows to satisfy the bound of the errors for other initial configurations. The controller parameters are $\mathbf{K} \equiv (d_i^{-1}, 0.6, 10, 0.1, 0.3, 0.01)$ and the upper bound of the errors are $E_{dist} = 0.1\ m$ and $E_\angle = 5°$. These parameters were chosen to obtain a smooth trajectory, fast response and velocity within the UGV's velocity limits (cf. Subsection 3.5.2). The kinematic constraints of the UGV are maximum velocity equal to $v_{max} = 2.5\ m/s$ and the minimum radius of curvature equal to $r_{c_{min}} = 2.83\ m$. Hence, using eq. (3.46) and (3.51), the minimum initial euclidean distance to the target is $d_i = 10.4\ m$.

In this first simulation, the desired final configuration is $(x_T, y_T, \theta_T) \equiv (10.4, 0, 0°)$ and $v_T = 0\ m/s$. The vehicle has an initial orientation θ_i equal to $80°$ and different initial distance equal to 7.4, 10.4 and $13.4\ m$. Fig. 3.21(a) shows the trajectory of the vehicle for these different initial distances. The distance and orientation errors are shown in Fig. 3.21(b). The Lyapunov function values are shown in Fig. 3.21(c).

Fig. 3.21(a) shows that the vehicle's trajectory towards its target depends on the initial distance. The upper bound of the distance and orientation errors are not satisfied when the initial distances ($7.4\ m$) is less than $d_i = 10.4\ m$ (dotted pink line). Fig. 3.21(b) shows that the system errors are bounded according to eq. (3.49) and (3.44) (black dashed line) and converge to zero (cf. Subsection 3.5.2).

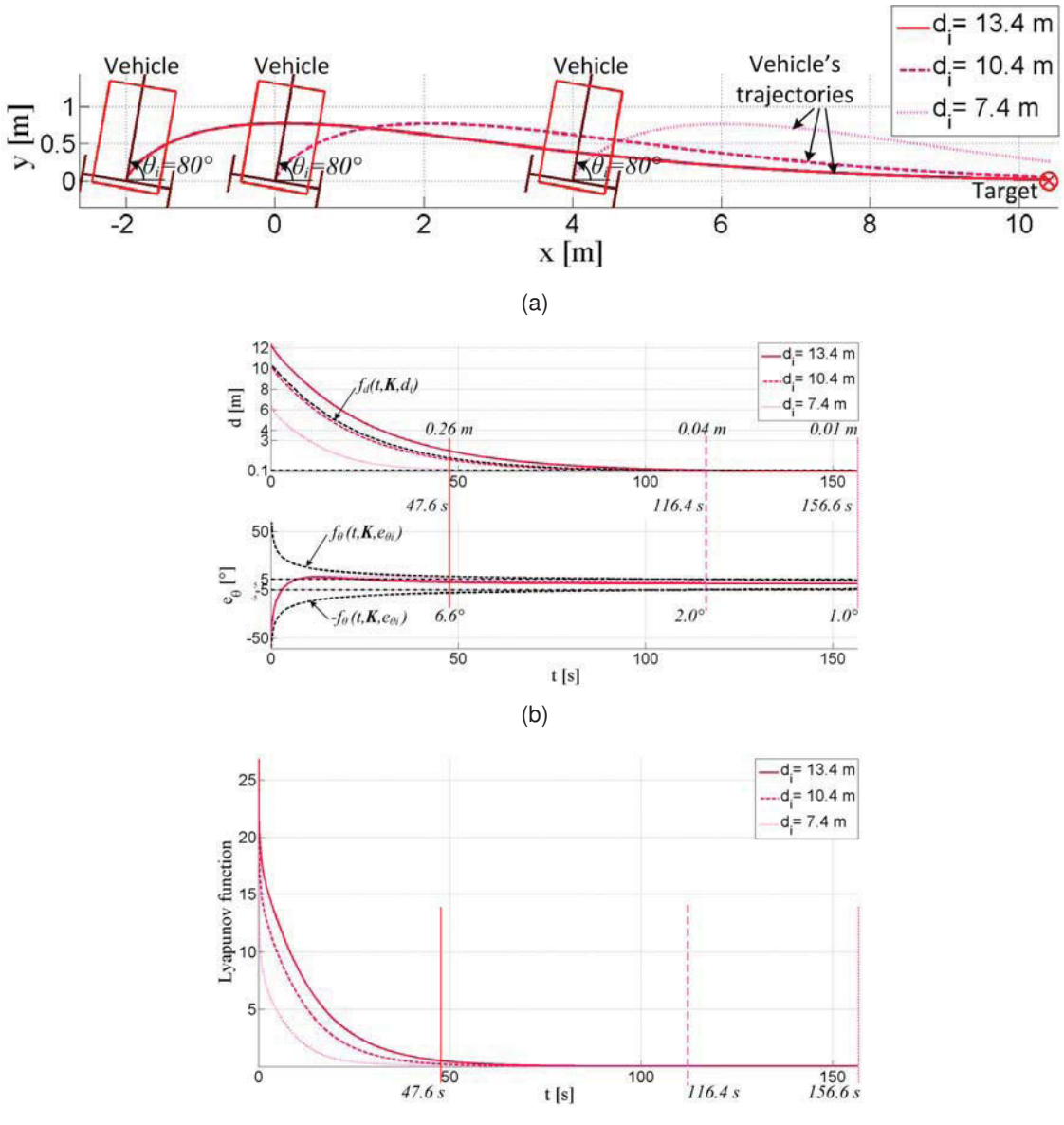


Figure 3.21: (a) UGV's trajectories, (b) their distance d and orientation errors e_θ and (c) their evolution of Lyapunov functions for several initial UGV's orientations.

In the second simulation, the desired final configuration is $(x_T, y_T, \theta_T) \equiv (15, 4, 0°)$ and $v_T = 1\ m/s$. Fig. 3.22(a) shows the trajectory of the vehicle for different initial orientations. The distance and orientation errors are shown in Fig. 3.22(b). The Lyapunov function values are shown in Fig. 3.23. Fig. 3.22(a) shows that the convergence of the system depends on the initial orientation error. Fig. 3.22(b) shows that the system errors satisfy the upper bound of the errors and converge to zero (cf. Subsection 3.5.2). Furthermore, the Lyapunov function shows asymptotic stability (cf. Fig. 3.23). Fig. 3.23(a) shows the three terms of the Lyapunov function eq. (3.32) where the first term is $0.5K_d d^2$, the second term is $0.5K_l d_l^2$ and the third term is $K_o[1 - \cos(e_\theta)]$. These figures show that the vehicle satisfies its constraints (velocity, acceleration and steering) according to the analysis presented in subsection 3.5.2.

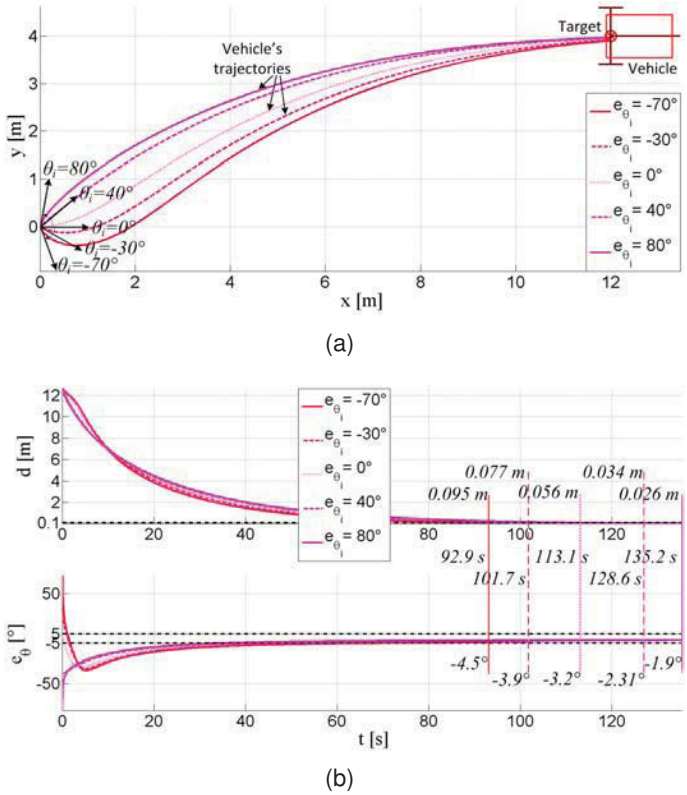


Figure 3.22: (a) UGV's trajectories and (b) its distance d and orientation errors e_θ for several initial UGV's orientations.

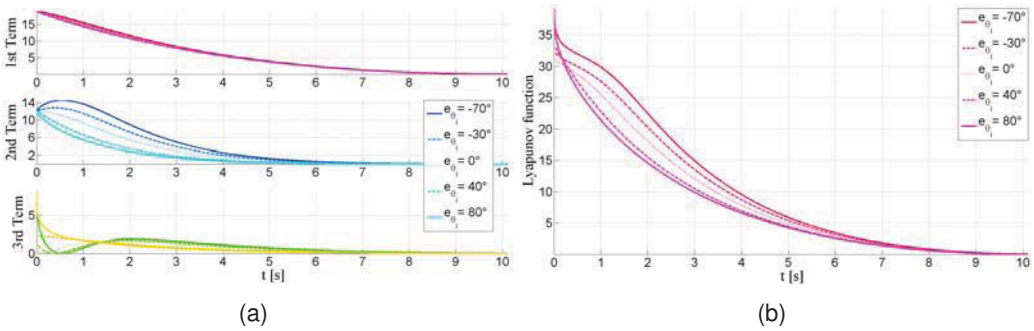


Figure 3.23: (a) Different terms of the Lyapunov function (cf. eq. (3.32)) and (b) Lyapunov function values for several initial orientations.

3.6/ EXPERIMENTAL VALIDATION OF THE OVERALL CONTROL ARCHITECTURE

This experiment focuses on the online UGV's navigation in cluttered environment using a LIDAR for online obstacle detection (cf. Fig. 3.24). The elliptical limit-cycle trajectories are applied for reactive obstacle avoidance using only position information and uncertain range data (cf. Appendix B.1). Each obstacle is surrounded by an appropriate ellipse (cf. Section 3.4.2) computed online using range data (cf. Fig. 3.25(a)). This experiment demonstrates the efficiency, feasibility, smoothness and safety of the proposed architecture and its elementary blocks (*Perception*, *Target reaching*, *Obstacle avoidance* and *Control law*) (cf. Section 3.3). This experiment can be found online[1].

In this experiment, the used UGV is an urban electric car named VIPALAB (cf. Appendix G.1). The VIPALAB has a range sensor (LIDAR) with the maximum detected range equal to $D_{max} = 10\ m$ (cf. Fig. 3.9). Moreover, the vehicle uses a combination of RTK-GPS and IMU using EKF to estimate its localization (current position and orientation) at a sample time of $T_s = 0.1\ s$ (cf. Appendix G.3). The controller parameters $\mathbf{K}$ are $\mathbf{K} \equiv (1, 2.2, 8, 0.1, 0, 01, 0.6)$. These parameters were chosen to obtain a safe and smooth trajectory, fast response and velocity value within the limits of the vehicle, which are $v_{max} = 2.5\ m/s$ and the minimum radius of curvature $r_{c_{min}} = 2.83\ m$.

Fig. 3.24 shows the cluttered environment with three static obstacles (two boxes and one static vehicle) and a VIPALAB. The red point is the assigned target. Some screenshots of the developed Graphical Data Interface for VIPALAB (GDI-VIPA) are shown in Fig. 3.25(a). The white line represent UGV's trajectory and the big red point is the assigned target. It can be noted the evolution of the online enclosing ellipse (orange circle) according to historic of the observed range data.

Fig. 3.25(b) shows the safe trajectory of the vehicle in the real cluttered environment where the parameters of the ellipse that enclose the obstacle are obtained online using the obstacle detection method [2, 3] (cf. Section 3.4.2.1). It can be observed that the

[1] http://maccs.univ-bpclermont.fr/index.php/Profiles/VilcaJM/NavigationVipalab.mp4

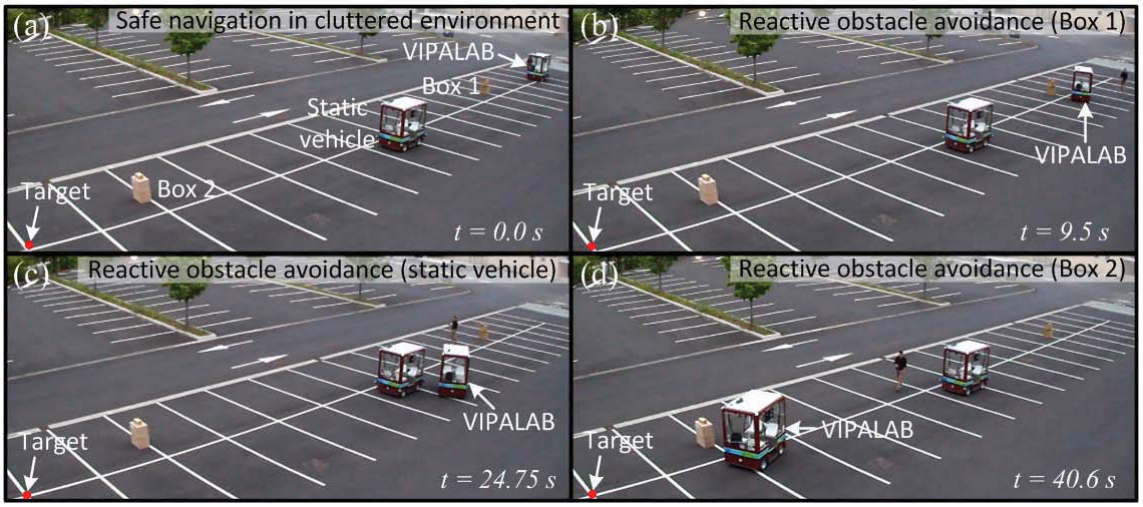
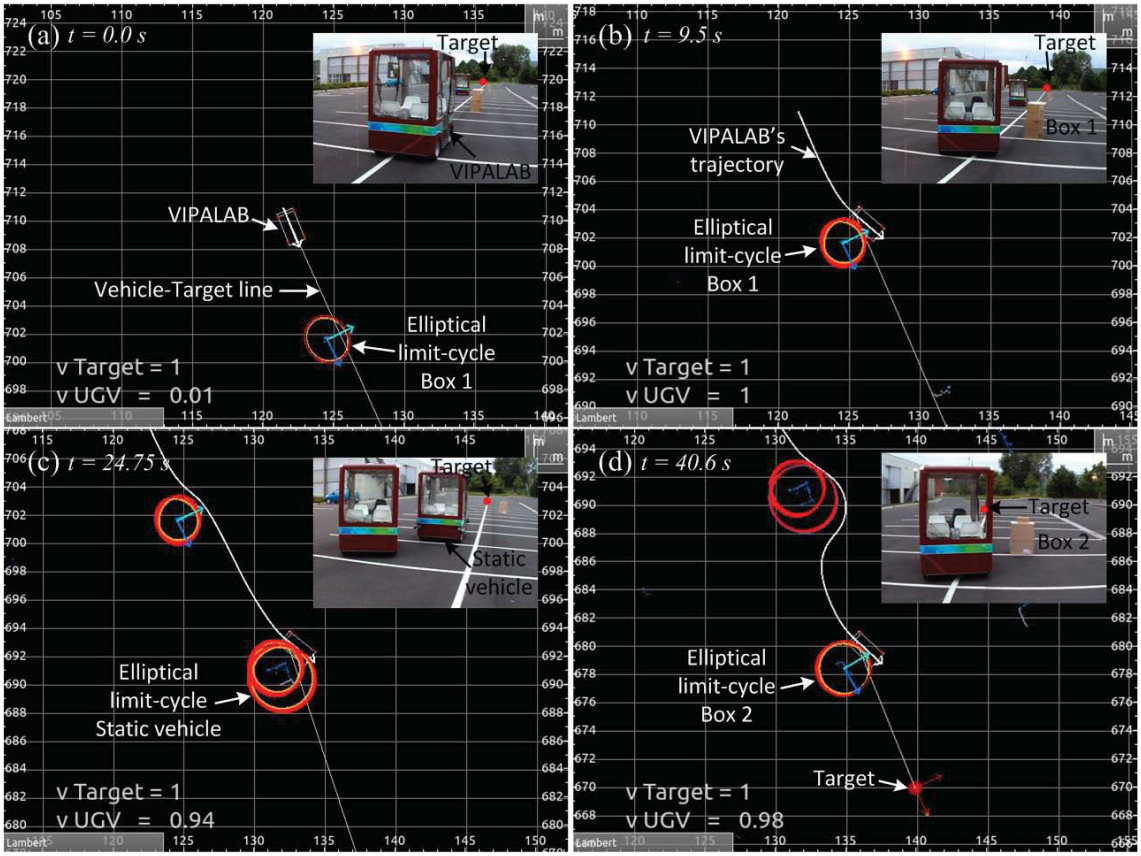
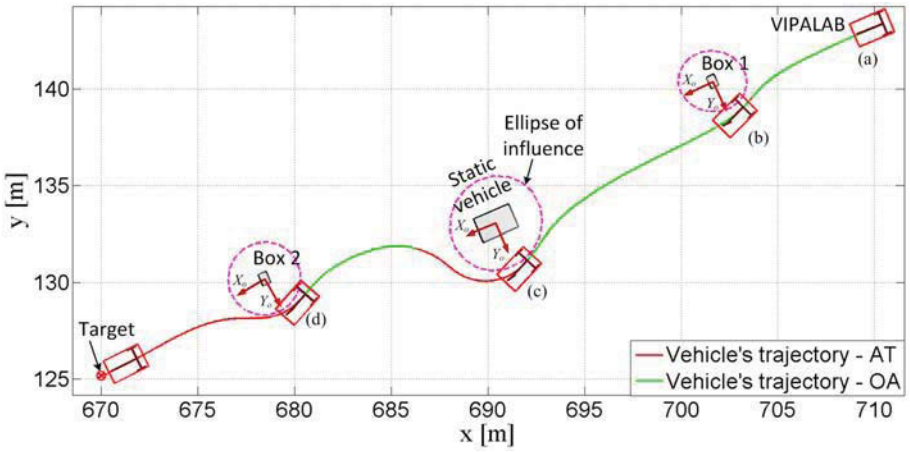


Figure 3.24: Safe and reactive VIPALAB's navigation while avoiding hinder obstacles.

vehicle avoids the obstacles with a smooth trajectory and converges to its target after avoiding three obstacles (two boxes and one static vehicle). This trajectory was obtained while using the reactive obstacle avoidance algorithm [293] (cf. Subsection 3.2) with the proposed control law (cf. 3.5) which takes its parameters from the online obstacle detection method (cf. Section 3.4).

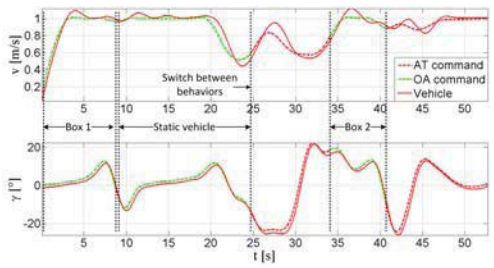


(a) GDI-VIPA during the safe navigation ("v UGV" and "v Target" are respectively the current velocities of the UGV and the target).

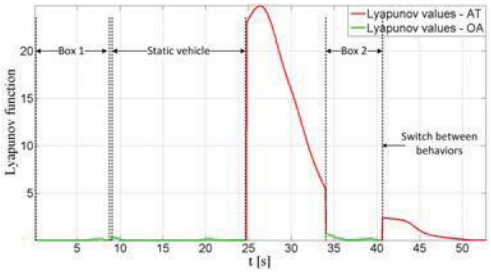


(b) Vehicle's trajectory using the proposed control architecture (AT: Attraction to the Target and OA: Obstacle Avoidance).

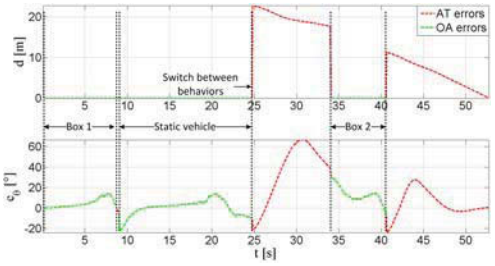
Figure 3.25: Validation of the safe navigation in cluttered environment.



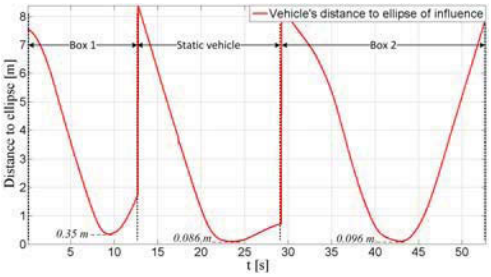
(a) Commands given by the control law.



(b) Evolution of Lyapunov function during the vehicle navigation.



(c) Position and orientation errors of the vehicle.



(d) Distance from the vehicle to the ellipse of influence E_{f_i}.

Figure 3.26: Experimental results of the safe navigation in cluttered environment (AT: Attraction to the Target and OA: Obstacle Avoidance).

Fig. 3.26(a) shows the output of the control law. It can be observed that the velocity decreases according to the distance to the obstacle (cf. Subsection 3.3.4) which allows to obtain a smooth and safe navigation. Fig. 3.26(b) shows the variation of the Lyapunov function V eq. (3.32) related to the control law (cf. Section 3.5) during the navigation task. This function decreases asymptotically which guarantees the stability of the system. Fig. 3.26(c) shows the controller parameters (position and orientation errors). It can be noted that the absolute value of errors converges towards zero after each behavioral switch. Therefore, the control law given by (3.23) and (3.24) remain always asymptotically stable.

Fig. 3.26(d) shows the minimum distance between the ellipse of influence (cf. Section 3.1) of each obstacle (obtained while knowing all the range data of the obstacles) and the vehicle's position along the limit-cycle trajectory (green line in Fig. 3.25(b)). The switching times between the obstacles in this figure were selected to show how the vehicle evolves along the elliptical limit-cycle. It can be noted that the UGV does never collide with any obstacle since there is not intersection with the ellipses of influence.

Therefore, the proposed control architecture is validated for real implementation to deal with cluttered environments.

3.7/ CONCLUSION

This chapter presented an overall control architecture to cope with the problem of navigation in a cluttered environment. The problem of obstacle avoidance is dealt with the use of online adaptive elliptic trajectory to perform smooth and safe UGV's navigation in a reactive way. These trajectories use the limit-cycle principle to obtain generic and flexible

navigation in very encumbered environments. This elliptic limit-cycle trajectory is obtained while using the proposed *Perception* block which deals with the uncertain range data to obtain the ellipse parameters of each obstacle. This method was demonstrated and implemented online to enclose all range data. Furthermore, a single control law suitable for the proposed control architecture is described. It takes into account the position and orientation set-points to improve the performance and the safety of UGV's navigation. This proposed control law was synthesized using a suitable Lyapunov function, which takes into account the position, the angle between the robot and the target, and its orientation with respect to set-points. Moreover, it enables static and dynamic target reaching. The stability of the overall control architecture was proved using a suitable Lyapunov function based on a new set of variables. The relation between controller parameters and the upper bound on distance and orientation errors at final time t_f allows to guarantee the safety of the UGV's trajectory towards its assigned target. Furthermore, different target-reaching methods from the literature were presented and their performances were compared with the proposed control law by simulations. These comparisons showed the interesting features of the proposed control law in term of stability and flexibility for different tasks. Experiments using a urban electric car validate the efficiency and the reliability of the proposed control architecture in cluttered environments.

4

SAFE AND FLEXIBLE AUTONOMOUS VEHICLE'S NAVIGATION USING SEQUENTIAL WAYPOINTS IN STRUCTURED ENVIRONMENTS

Different strategies for autonomous navigation in structured environments have been proposed in the literature [293, 324, 327] (cf. Fig. 4.1). The most popular approaches are based on the following of a pre-defined reference trajectory [37, 222, 319]. These approaches link the control to a reference trajectory which can be obtained with a combination of path planning and trajectory generation techniques [221]

Typically, to obtain the reference path to be followed by the vehicle, arc-lines, B-splines or polynomial equations are used to interpolate points/waypoints [214, 229, 254, 327]. This safe path can be a time-parametrized path or a path without temporal reference [221]. The time-parametrized path can take into account different UGV's constraints and environmental characteristics [192]. Nevertheless, obtaining this path requires more computational time than obtaining path without temporal reference [253]. Most of the control laws linked to a reference path are dedicated to either *trajectory tracking* (to track a time-parametrized reference [318]) or *path following* (to follow a path without explicit temporal references [77]).

In [324] a feasible path is obtained using a polynomial curvature spiral. In [319], the trajectory generation method generates a smooth path considering the kinodynamic constraints of the vehicle. In [266] straights line paths, defined by the position and orientation

Figure 4.1: Autonomous navigation of a UGV in an urban environment (Clermont-Ferrand, France).

of a single waypoint, are considered. In this case, the orientation of the previous waypoint is not taken into account to simplify the implementation of the control law. In [254], trajectories are built using user assigned points and interpolation functions such as cubic splines, trigonometric splines and clothoids. Moreover, velocity profiles along the trajectory are specified to improve the passengers comfort which is related to the acceleration. Nevertheless, trajectory generation presents some drawbacks, such as the necessity of a specific planning method, the proof of guarantee of continuity between different segments of the trajectory and the complexity for replanning.

A few works in the literature propose to use only specific set of waypoints in the environment to lead the robot toward its final objective. In [69], the authors propose a navigation strategy via assigned static points for a unicycle robot. This strategy does not allow accurate navigation since the kinematic constraints of the robot (maximum velocity and steering), the orientation error and the velocity profile of the robot when it reaches the assigned point are not considered. Harmonic Potential Field (HPF) is used to guide an unmanned aerial vehicle (UAV) to a global waypoint with a position and a direction of arrival in [332]. The author proposes a virtual velocity field which allows to consider the UAV model. Each vector component of the field is treated as an intermediate waypoint and the robot must comply with it in order to reach the global waypoint. Nonetheless, HPF requires a complex mathematical modeling for different shapes or dimension of the obstacles in the environment.

In this chapter, the proposed navigation strategy which avoids the pre-generation of any specific reference trajectory is presented. Vehicle movements are obtained according to the proposed control law while considering vehicle kinematic constraints and sequential waypoints to reach (defined by its position, orientation and velocity) (cf. Section 3.3.3). The vehicle can thus perform different movements between waypoints without the necessity of replanning any reference trajectory. Moreover, it can also add or change the location of the successive waypoints according to the environment configuration or to the task to achieve. Thus, this strategy allows flexible navigation while taking into account appropriate waypoints suitably placed in the environment (cf. Section 4.3).

Next section describes the motivation and the considered structured environment for UGV's navigation. The proposed complete framework for autonomous vehicle navigation through successive waypoints is described and validated by simulation in Section 4.2. Furthermore, different algorithms from the literature to obtain the set of waypoints in the environment are detailed in Section 4.3. This sections also proposes an elementary waypoint selection method from a reference trajectory to perform safe and smooth trajectories and two methods for the selection of waypoint configurations in a well-known environment using a multi-criteria optimization techniques. The proposed algorithms are validated through several simulations to demonstrate their advantages, safety, reliability and flexibility. An extension to multi-robot system is also proposed in Subsection 4.3.4.5. In Section 4.4, our navigation strategy is implemented and experimented in a real urban electric vehicle. Finally, Section 4.5 provides a conclusion of the chapter.

4.1/ PROBLEM STATEMENT

Typically, a human driver reactively guides his vehicle while performing a smooth trajectory within the roads limits until reaching the defined goal. To achieve similar behavior with a UGV, a target assignment strategy to enable vehicle navigation through successive

waypoints in the environment is presented.

An important condition in the field of autonomous vehicles is to ensure safe and flexible vehicle navigation in a structured environment (cf. Fig. 4.1 and 4.2). In this work, safe navigation consists in not crossing over the road limits and bumping into obstacles while respecting the physical constraints of the vehicle. Flexible navigation consists in allowing different possible movements to achieve the task, while guaranteeing a smooth trajectory of the vehicle. The main idea of the proposed work is to guarantee both criteria simultaneously. The analysis of the control law presented in Subsection 3.5.2 allows the UGV to perform safe navigation within a certain boundary.

In this section, it is shown that it is not always important to follow with a high fidelity an imposed path for a vehicle, specifically in open or low-constrained environments. It will be demonstrated that only few waypoints will be sufficient to guarantee safe and flexible UGV navigation. A target assignment strategy is proposed to perform autonomous navigation through pre-defined waypoints (cf. Section 4.2). Moreover, it is also demonstrated that if we increase the number of these waypoints, the robot control performs as if we had applied common trajectory tracking control.

In what follows, the considered scenario for the UGV's navigation through sequential waypoints is presented (cf. Fig. 4.2):

- The environment is known throughout a map, containing the position of all obstacles and road dimensions.

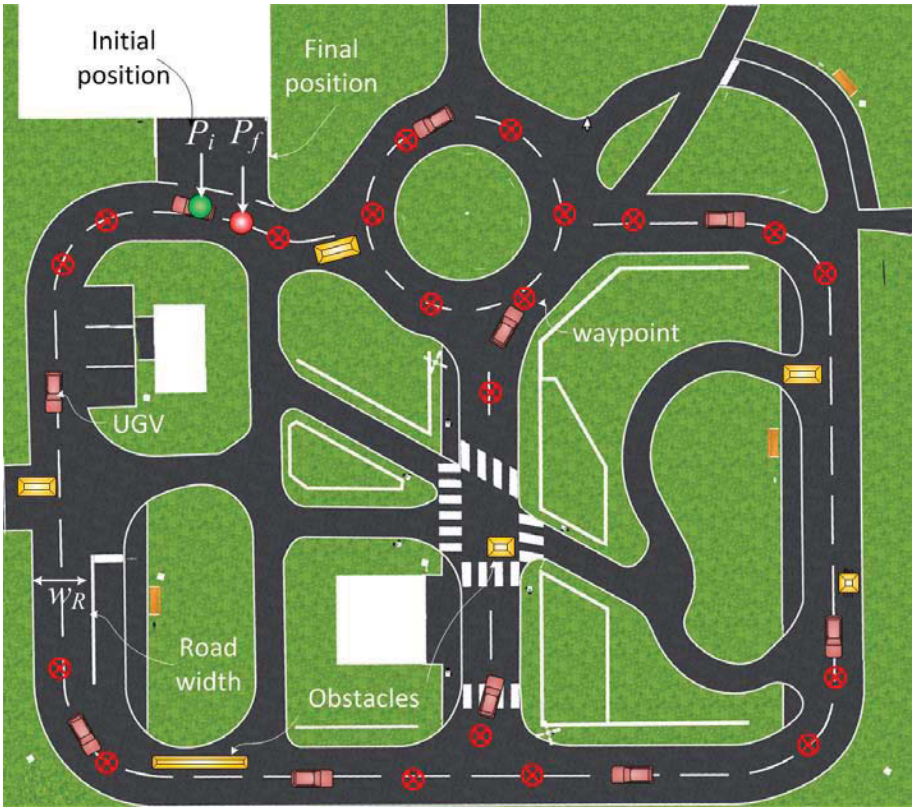


Figure 4.2: Nominal scenario with a road map and the task to achieve by the vehicle in its environment.

- The kinematic/dynamic model of the vehicle is known (with potential uncertainties).
- The vehicle starts at the initial pose P_i and it has to reach the final pose P_f (in certain cases, $P_i = P_f$).

As presented in the beginning of Chapter 4 and according to the described scenario, a safe reference path in static environment can be obtained by different algorithms such as Voronoï diagram [46], potential fields [18] or others [221]. In this work, specific key positions should be defined in the static environment, which are named *waypoints* (cf. Fig. 4.2). A waypoint corresponds to a specific key configurations $(x_{p_i}, y_{p_i}, \theta_{p_i}, v_{p_i})$ (where $(x_{p_i}, y_{p_i})^T$, θ_{p_i} and v_{p_i} denote respectively the position, the orientation and the velocity of the waypoint p_i) in the environment as given in Fig. 4.3 (cf. Section 4.2).

UGV's navigation using only waypoints allows to avoid any path/trajectory planning which could be time-consuming and complex, mainly in cluttered and dynamic environment. Moreover, this kind of navigation does not require the pose of the closest point to any trajectory (w.r.t. the robot configuration) and/or the value of the curvature at this point [324]. Consequently, the navigation problem is simplified to a target (waypoint) reaching problem, i.e, the UGV is guided by waypoints (cf. Fig. 4.3) instead of following a specific fixed path.

4.2/ NAVIGATION STRATEGY BASED ON SEQUENTIAL TARGET REACHING

The proposed strategy uses a sequence of sorted waypoints suitably disposed in the environment [4, 8]. This sequence is obtained by a method which selects the optimal set of waypoints to perform a safe vehicle navigation in structured environment (cf. Section 4.3). In this section, the set of waypoints is assumed known to focus on the navigation strategy (cf. Fig. 4.4). Fig. 4.3 shows a set of successive waypoints. D_j is the Euclidean distance between the waypoints p_{i-1} and p_i. For simplicity, the orientation of the waypoint

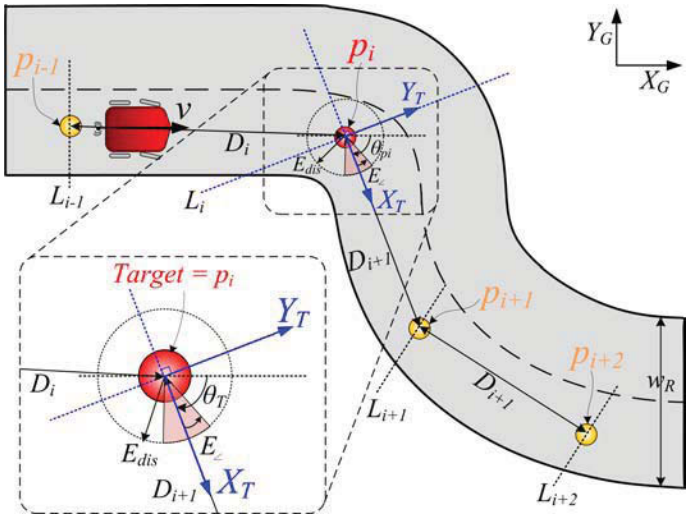


Figure 4.3: Description of waypoints and target assignment.

θ_{p_i} is represented as the orientation of the line that joins p_i and p_{i+1}:

$$\theta_{p_i} = \arctan\left(\frac{y_{p_{i+1}} - y_{p_j}}{x_{p_{i+1}} - x_{p_i}}\right) \tag{4.1}$$

To navigate between successive waypoints (e.g. p_{i-1} and p_i), the distance d of the vehicle to the target (current waypoints to be reached) and the error e_θ between the orientation of the vehicle and the target are used. Their maximum values (E_{dis} and $E_\angle$ respectively (cf. Fig. 4.3)) are imposed for the current waypoint p_i to be reached (cf. Subsection 3.5.2). These values are related to the inaccuracies of the vehicle localization and/or the performance of the used control law. The maximum authorized values allow to keep a reliable vehicle control towards the target p_i (cf. Fig. 4.3) while guaranteeing the appropriate vehicle configuration to reach the next target p_{i+1}.

4.2.1/ SEQUENTIAL TARGET ASSIGMENT

Fig. 4.4 shows the extension of control architecture, proposed in section 3.3, for the navigation through successive waypoints. At this aim, a new block, *Sequential target assignment* is added. The proposed navigation strategy can be extended easily in order to deal with dynamic environments, notably using the limit-cycle approach [293], [3] (cf. section 3.1). This obstacle avoidance controller allows to modify locally the movement of the robot to avoid dangerous static or dynamic obstacles and to come back to its initial plan.

The flowchart of the proposed strategy to assign at each sample time the appropriate waypoint is shown in Fig. 4.5. The parameters of the control law (cf. Subsection 3.5) enable the vehicle to reach the next waypoint with an assigned velocity (3.23) (this velocity can be different from zero) while ensuring that the vehicle trajectory is smooth and always within the road limits (cf. Section 4.2.3).

The current waypoint is named the *Target*. The initial target is the first waypoint from the set of sorted waypoints. The error conditions (E_{dis} and $E_\angle$) are used to switch to the next waypoint in the list, when the vehicle position enters in the circle with a radius equal to E_{dis} and center (x_{p_i}, y_{p_i}). If the vehicle does not satisfy the distance and orientation error conditions w.r.t. the current waypoint p_i then the perpendicular line L_i (Y_T axis) to the $\overline{p_i p_{i+1}}$ line at the current waypoint is used to switch to the next waypoint when the vehicle crosses the line L_i, i.e., $x^T \geq 0$ where x^T is the coordinate of the vehicle in the local Target frame $X_T Y_T$ (cf. Fig. 4.3). The current target is updated with this next waypoint which should not intersect any obstacle (otherwise, a new next waypoint is selected until this condition is satisfied) (cf. Fig. 4.5). Finally, the vehicle starts the movement to reach this new current target.

4.2.2/ SMOOTH SWITCHING BETWEEN WAYPOINTS

When the vehicle switches from one waypoint to another (e.g., from p_j to p_{j+1} as depicted in Fig. 4.3), the value of controller variables $\mathbf{C}_v = (e_x, e_y, e_\theta, e_{RT}, v_T, r_{cT})$ (cf. Section 3.5) can change abruptly. These hard switches could induce, in certain situations, the actuators jerk (v and γ from eq. 3.23 on page 62 and 3.24). This aspect could generate in certain applications, such transportation task, the discomfort of passengers.

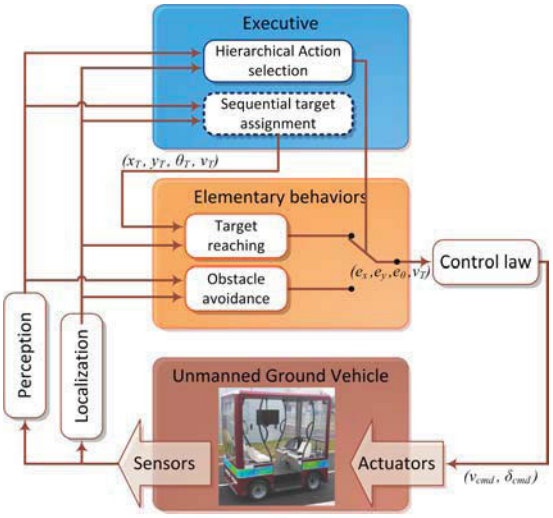


Figure 4.4: Proposed control architecture for navigation through waypoints.

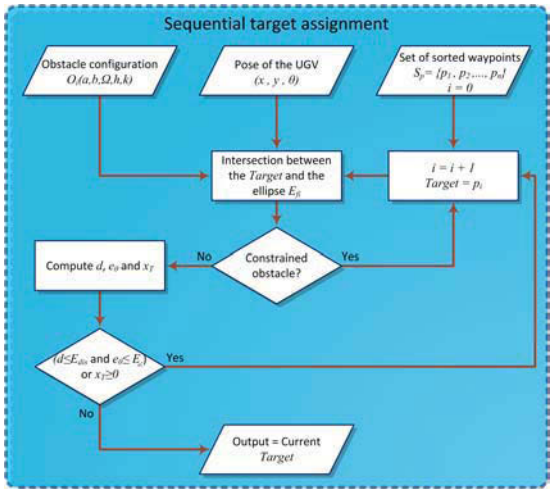


Figure 4.5: Flowchart of the Sequential target assignment.

This subsection proposes to avoid this hard switch by introducing a smooth evolution of this control variables for a certain amount of movement distance d_s (smoothness distance) without disturbing the safe vehicle's navigation [4]. This distance depends on initial distance ($\%\ of\ d_i$) separating the vehicle to the next waypoint p_{j+1}. A Sigmoid function is applied to the controller variables $\mathbf{C}_v$ along of d_s. The new Smooth Virtual Controller variables ($\mathbf{SVC}_v$) are designed according to the covered distance $d_c = d_i - d$, where d is the current distance to the current target (cf. Fig. 4.6). The $\mathbf{SVC}_v$ function is given by:

$$\mathbf{SVC}_v(d_c) = \mathbf{C}_{vi} + \frac{(\mathbf{C}_v - \mathbf{C}_{vi})}{1 + e^{-a_s(d_c - d_0)}} \tag{4.2}$$

where $\mathbf{C}_{vi}$ and $\mathbf{C}_v$ are respectively the initial and current values of the controller variables. For example, for e_x, an element of $\mathbf{C}_v$, when the target switches from p_j to p_{j+1}, e_{xi} is the value before to switch to p_{j+1} and e_x is the current error w.r.t. the waypoint p_{j+1}; d_0 is the value where the function has a half of its current value and a_s is a constant value related to the slope of the sigmoid function. It is designed to attain the effective value ($\mathbf{SVC}_v \approx \mathbf{C}_v$) when $d_c = d_s$ (cf. Fig. 4.6).

Before to detail the waypoints selection strategies, let us first evaluate the performance of the navigation through successive waypoints.

4.2.3/ Simulation evaluation of the navigation strategy

This section presents simulations to demonstrate the efficiency of navigation through successive waypoints. As mentioned in Section 3.6, the physical parameters of the urban vehicle VIPALAB (cf. Section G.1), modeled using the tricycle kinematics (cf. Section 3.3.1), and the proposed control law (cf. Section 3.5) were considered.

The values of the controller parameters (cf. eq. 3.25 on page 62 and 3.26 on page 62) are $\mathbf{K} = (1/d_i, 1.8, 8, 0.15, 0.6, 0.01)$ (d_i is the initial distance to the target). These parameters were chosen to obtain a safe and smooth trajectory (even during the target switching), fast response and velocity value within the limit of the vehicle, which are minimum movement velocity $v_{min} = 0.1\ m/s$, maximum velocity $v_{max} = 2.5\ m/s$, maximum steering angle $\gamma_{max} =$

$\pm 23^{\circ}$ (or minimum radius of curvature $r_{c_{min}} = 2.83\ m$) and maximum linear acceleration $1.0\ m/s^2$. The sample time is $T_s = 0.01\ s$

Firstly, the case of successive waypoints reaching through a set of waypoints is analyzed. Two set of waypoints selected from a reference trajectory are used, one set has a distance between waypoints equal to $2\ m$ and the other equal to $4\ m$ (cf. Fig. 4.7(a)). Figures 4.7(a) and 4.7(b) respectively, show the vehicle trajectories and lateral and angular errors w.r.t. the reference trajectory for two set of waypoints. It can be noted that the obtained vehicle trajectories are close to the reference trajectory; and as expected, the lateral and angular errors are smaller when the fixed distance between the waypoints decreases. Therefore, the proposed navigation strategy and control law permits also to the vehicle to perform accurate trajectory tracking behavior if the waypoints are close enough.

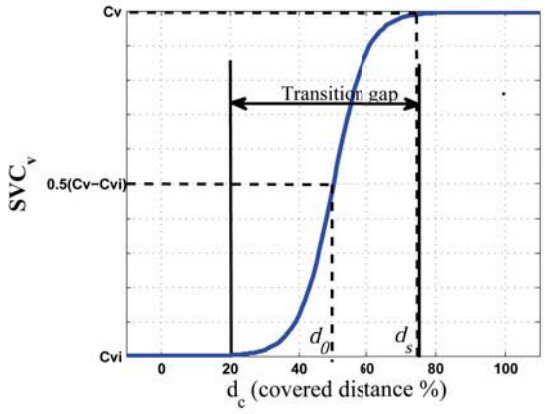
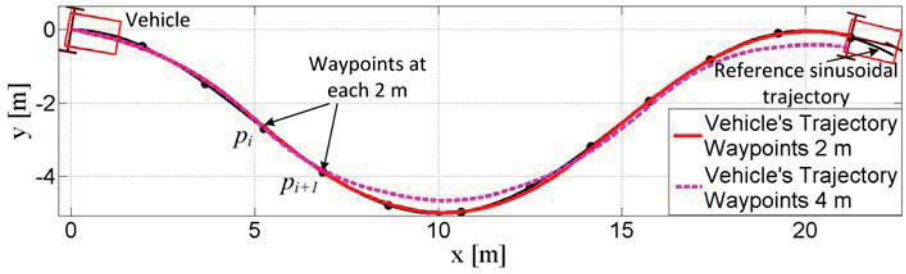
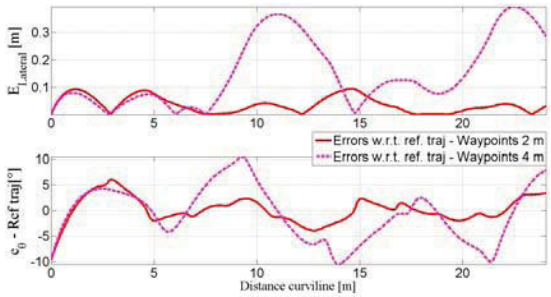


Figure 4.6: Evolution of the $\mathbf{SVC}_v$ used to ensure smooth control when the waypoint switching occurs.



(a) Vehicle's trajectories.



(b) Errors w.r.t. reference trajectory.

Figure 4.7: Simulation results for different distances between waypoints.

Secondly, the performance (safety, smoothness and convergence) of the control law to reach a desired target (static or dynamic) located at different distances. For each simulation the UGV starts at the same configuration but with different values of waypoints' velocities.

Fig. 4.8 shows the trajectories of the UGV for different static waypoints ($p_i,\ i = 1,\ldots,6$) and for a dynamic target p_d (sinusoidal trajectory). The static waypoints are positioned at different initial distances d_i and orientation angles between them ($45°$ until p_5 and $0°$ for p_6). The velocity profile of the waypoints for each simulation are $v_p = 0.1,\ 0.5$ and $1.0\ m/s$ respectively. It can be noted that the vehicle converges to each waypoint (static and dynamic), located in different positions and with different set of velocities. The dynamic target starts its movement when the vehicle reaches the last static waypoint p_6.

Fig. 4.9(a) shows the values of errors d and e_θ for all waypoints to reach. For static waypoints ($p_i,\ i = 1,\ldots,6$), the obtained values of errors just before to switch from waypoint p_i to p_{i+1} are shown. For dynamic target, the evolution of $d(t)$ and $e_\theta(t)$ during all tracking phase are shown. It is observed that the distance and orientation errors of static waypoints depend on initial UGV's configuration (pose and velocity). The errors increase when the static waypoints are closer. Moreover, for static waypoints where the profile velocity is the desired vehicle velocity at target position, the different profiles velocities have similar errors. It occurs in static waypoints because the proposed control law (angle steering) relies only on the dynamic of pose errors between the vehicle and the static waypoint (cf. eq. (3.24 on page 62) and (3.26 on page 62)) (the waypoint's velocity is a set-point for the vehicle when it reaches the waypoint's pose (cf. eq. (3.23 on page 62)) and the controller parameters $\mathbf{K}$ are tuning according to same initial distance d_i to the waypoint. Obviously, for dynamic target, the small target profile velocity has a faster convergence to zero.

The convergence of the Lyapunov function (3.32) is shown in Fig. 4.9(b) (cf. Subsection 3.5.1 on page 63) when the vehicle starts at the same initial position but with different initial orientations to the waypoint p_1 and while taking $v_p = 0.5\ m/s$. The use of Sigmoid function (cf. Subsection 4.2.2) is observed in the Lyapunov function and vehicle commands (velocity and steering angle) (cf. Fig. 4.9(b) and 4.9(c)) for the static waypoint with profile velocity of $v_p = 0.5\ m/s$. It can be noted that the Sigmoid function contributes to avoid peaks at the transition time and to obtain thus smooth vehicle commands while maintaining the stability of the control.

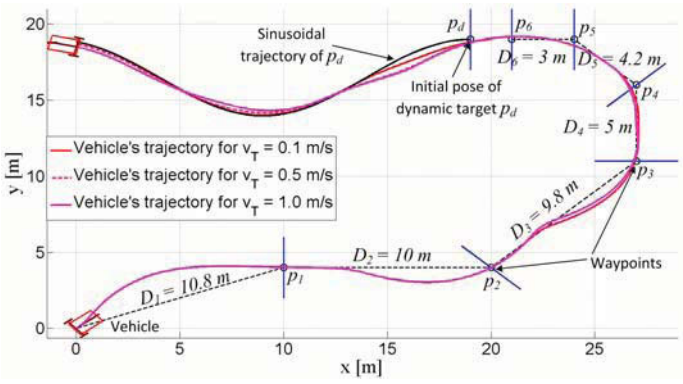
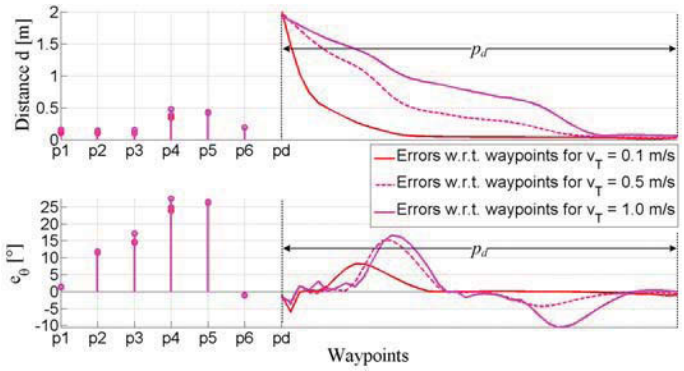
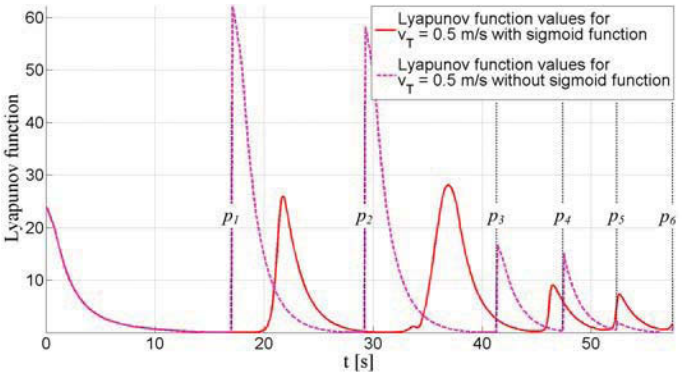


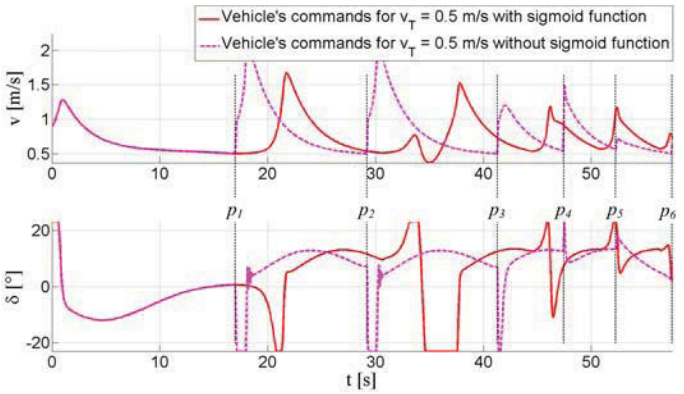
Figure 4.8: Trajectories of the vehicle for several waypoints' velocities.



(a) Distance and orientation errors for several waypoints' velocities.



(b) Values of the Lyapunov function for $v_p = 0.5\ m/s$ with and without the adaptive Sigmoid function.



(c) Control commands with and without the adaptive Sigmoid for smooth transition (**SVC**$_v$.

Figure 4.9: Simulation results of the navigation through sequential waypoints

4.3/ SELECTION OF WAYPOINT CONFIGURATIONS IN STRUCTURED ENVIRONMENTS

Once the proposed navigation strategy based on sequential target reaching is validated, in term of control stability and smoothness (cf. Section 4.2.3), let us address , in this section, the methods to obtain the most appropriate configuration (number, poses, etc.) of these waypoints in the environment. The aim is to ensure in all cases the navigational

safety, but in addition, the smoothness and rapidity of the navigation can be taken into account.

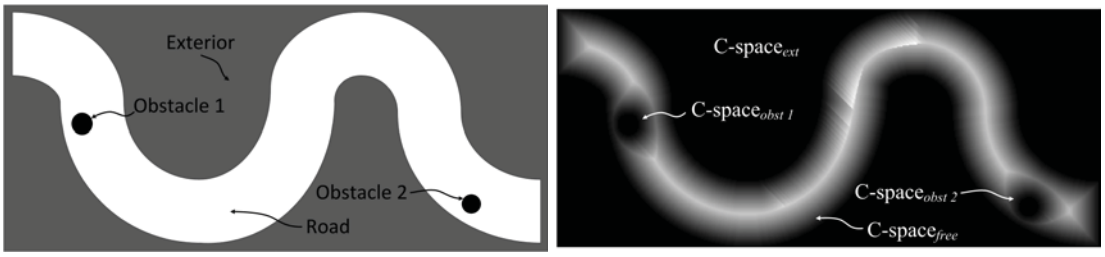
Waypoint selection consists of obtaining minimum number of points (waypoints) on the road to be successively reached by the vehicle to perform a safe navigation. These waypoints are selected considering a safe position on the road (as far as possible from the road limits) and the reliability of the obtained vehicle trajectory (smooth changes between the successive points). Before to present the proposed methods for the selection of waypoint configuration, in what follows, the state of the art related to waypoint planning is given.

4.3.1/ STATE OF THE ART

Diverse algorithms can be used to obtain these waypoints such as A*, D* [192], Rapidly Random Tree (RRT) [253], Sparse A^* Search (SAS) [123] and others [192, 221]. A useful tools for these algorithm is the Configuration space (C-space) which is the space of all possible configuration of the robot [244]. The C-space enables the identification of the safe area where the vehicle can navigate without a collision risk (free space C-space$_{free}$). The C-space is also used to compute the minimum distance to C-space$_{obst}$ (obstacle) and/or C-space$_{ext}$ (road boundaries). Fig. 4.10 shows the C-space and its Voronoï diagram [46] in gray scale w.r.t. the distance to the closest C-space$_{obst}$ and C-space$_{ext}$ (the whitest area represents the safest area).

Typically, algorithms based on grid map (e.g., A* or D*) produce the shortest path by optimization of a criterion such as the distance to the goal, distance to the risk area, etc. [192]. The algorithm begins generally at the final cell (final position) and traverses the cell's neighbors until to reach the initial position. The cost of traveling through the neighbor is added to the total cost, the neighbor with the lowest total cost is selected, and so on. The process terminates once the initial position is reached. The path is given through the cell positions of the grid map while backtracking the cells which have the lower path cost. Sometimes, a polynomial interpolation is used to obtain a smooth path [229]. In [259], the authors present an A* algorithm using clothoid trajectories assuming constant velocity along them. Therefore, appropriate waypoints can be selected from this shortest path while only considering the cells where an orientation change occurs (w.r.t. its predecessor). Nonetheless, this algorithm does not consider neither former/initial vehicle orientation nor its kinematic constraints.

Instead of using grid map, it is possible also to obtain safe, feasible and smooth path using



(a) (b)

Figure 4.10: (a) Road scheme and (b) its C-space with its Voronoï diagram.

expanding tree algorithms (e.g., RRT, RRT* or SAS [123, 221, 253, 301]). This could be done by providing to the vehicle's model the commands to reach the successive selected nodes until its goal [123, 221, 253]. The basic process of RRT consists in selecting, at each sample time, a random node q_{random} in the C-space$_{free}$. This selection considers generally only position $q_{random} = (x_{random}, y_{random})$ without any *a priori* UGV's orientation [221]. Then, the commands (discrete values) are applied to vehicle (from its current position and orientation) during a constant time t_{exp}. The vehicle's model and constant commands allow to predict the final vehicle position at the end of t_{exp}. The commands that produce the closest position q_{chosen} (a node which optimizes a dedicated task criterion [291]) to current random node q_{random} are selected and stored with q_{choose}. A new expansion starts until to reach q_{random} or to select a new random node q^{new}_{random}. Therefore, the waypoints can be selected, as in the case of grid map, while only considering the nodes where an orientation change occurs (w.r.t. its predecessor node). Algorithms based on RRT are very useful for motion planning because they can provide the commands (based on the kinematic/dynamic model of the vehicle) to reach the desired final configuration [253, 291]. Moreover, RRT avoids the use of grid maps that can increase the computation time for large environment. In [123], the authors use the expanding tree for trajectory planning introducing different constraints such as maximum turning angle and route distance. Nevertheless, this method does not consider neither the vehicle movements along the trajectory nor localization uncertainties. In [300], sequential composition of controllers (e.g., go to the landmark and wall following controller) are used to generate valid UGV's states q_{choose} to the navigation function. This approach avoids to find a single globally attractive control law and allows to use some additional sensing capability of the robot when the landmark is unreachable (e.g. GPS-denied area). However, the obtained navigation function has a complex computational processing. The most important drawbacks of expanding tree algorithms are the slow convergence to cover all space until to reach the goal and that in most cases these algorithms do not provide the shortest path since the nodes are randomly selected [292]. Furthermore, it is important to underline that in RRT the control commands are maintained during a certain time, whereas in this thesis the UGV's movement takes into account the definition of the proposed control law (cf. Section 3.5) in addition to the UGV's model (cf. Subsection 4.3.3.2). A comparison with RRT and Voronoï approaches is shown in Subsection 4.3.4.2.

Furthermore, a method to obtain the waypoints was proposed in [339]. The authors use agent's observation and the geometric characteristics of the environment to select the waypoints. These waypoints can be used to reduce the planning time of existing planners. However, the method is based only on the position; the orientation and UGV's model are not taken into account.

In this section, the proposed control architecture for navigation through successive waypoints (cf. Section 4.2) is extended to deal with vehicle planning/set-points definition. At this aim, a new block in the Planning layer, *Waypoint configuration selection*, is added. A new efficient and flexible algorithm to obtain the optimal set of waypoint configurations in the environment is proposed. Combining multi-criteria optimization and expanding tree allows safe, smooth and fast UGV's navigation (cf. Subsection 4.3.3.2). The method based on expanding tree to obtain the optimal waypoint configuration in a structured environment (cf. Fig. 4.2). This method allows to consider constraints such as the kinematic model and the used control law. Criteria to optimize are related to the kinematic constraints of the vehicle (non-holonomy, maximum velocity and steering angle) and localization uncertainties. To highlight the advantages and flexibility of the novel method,

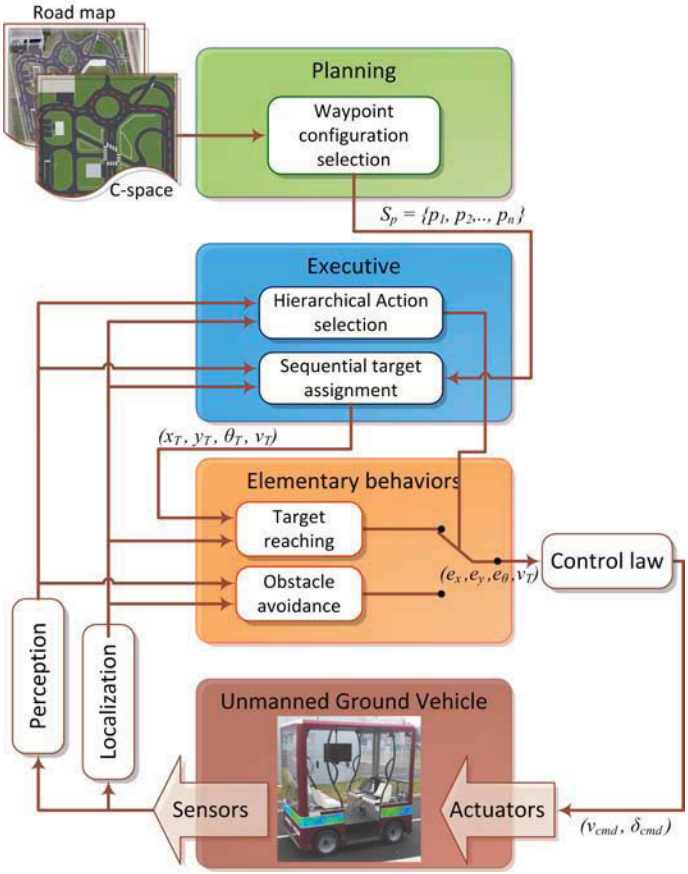


Figure 4.11: Planning block (green block) added to the control architecture for navigation through waypoints.

a comparison with another proposed method, based on the commonly used grid map, is presented (cf. Subsection 4.3.3.1). The method based on grid map algorithm considers the vehicle as one cell without constraints and it can move only through the cells of the grid map.

4.3.2/ PRIMARY METHOD FOR THE SELECTION OF WAYPOINT CONFIGURATIONS FROM A REFERENCE TRAJECTORY

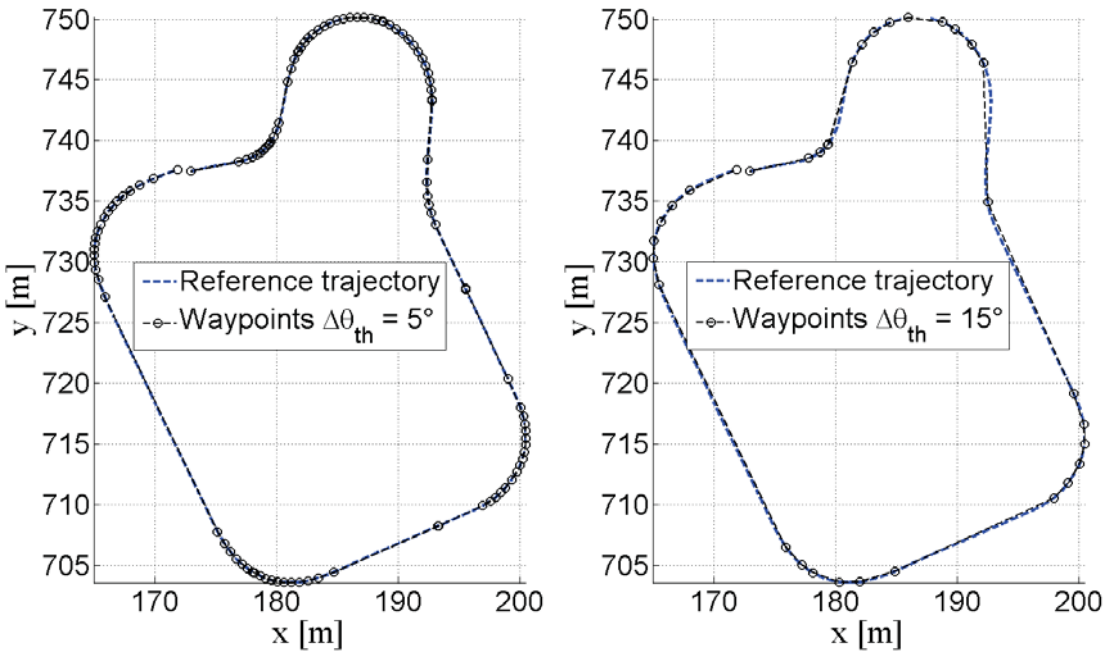
This method can be applied when a safe reference trajectory is already defined. This reference path could be obtained by different algorithms such as Voronoï diagram [46], potential fields [18] and others [221], or by using a recorded vehicle trajectory [4]. The aim of the proposed method is to select an appropriate number of waypoints using this reference path (cf. Algorithm 1) to apply the navigation strategy based on sequential target reaching (cf. Section 4.2). Different criteria can be considered to obtain the minimum number of straight lines that closely fit the reference path. Criteria such as the euclidean or curvilinear distance, orientation or radius of curvature between waypoints can be used to fix the desired waypoints on the path. The discretized reference path **r** is composed of sorted pose $r_i = (x_{r_i}, y_{r_i}, \theta_{r_i})$; where θ_{r_i} is the tangential orientation. The minimum number of straight line segments over the defined path is then computed while considering a constant threshold $\Delta\theta_{th}$ for the orientation variation along the path $\Delta\theta$ (cf. Algorithm 1).

Algorithm 1 Waypoint selection based on existing reference path

Require: Reference path $\mathbf{r} = (\mathbf{x}_r, \mathbf{y}_r)$ and $\Delta\theta_{th} \in \mathbb{R}^+$
Ensure: Set of waypoints S_p

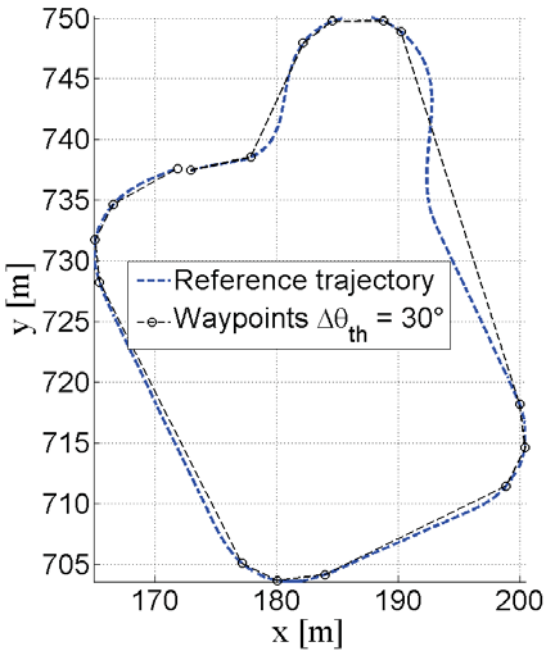
```
Init j = 0, p_j = r_0 (initial pose of r)
for r_i ∈ r with i ≥ 1 (sorted set of path points) do
    Compute Δθ = |θ_{r_i} − θ_{p_j}|        ▷ Current path point compares its orientation w.r.t. waypoint
    if Δθ ≥ Δθ_th then
        if r_{i−1} ∉ S_p then                ▷ Check if predecessor path point belongs to S_p
            Add predecessor path point r_{i−1} to S_p
        end if
        Add predecessor path point r_i to S_p
        j = j + 1
    end if
end for
```

Fig. 4.12 shows the obtained waypoints using Algorithm 1 with $\Delta\theta_{th} = 5°$ ($n_p = 113$), $15°$ ($n_p = 34$) and $30°$ ($n_p = 16$) respectively. Obviously, the switch between waypoints is smoother with a small value for $\Delta\theta_{th}$. Nevertheless, adding this step of pre-defined path planning (with all its possible drawbacks (cf. section 4.3.4)) before obtaining the set of waypoints, restricts considerably the C-space$_{free}$ to only a curvilinear line. Thus, the optimality of the obtained set of Waypoints is not guaranteed. Next section focus on optimization methods for the selection of waypoint configurations in any environment.



(a) $n_p = 113$ (b) $n_p = 34$

Figure 4.12: Example of waypoint selection based on a reference path and Algorithm 1.



(c) $n_p = 16$

Figure 4.12: Example of waypoint selection based on a reference path and Algorithm 1 (continuation (cont.)).

4.3.3/ Optimal Multi-criteria Waypoint Selection (OMWS)

This section is dedicated to the selection of the discrete waypoints in structured environment (cf. Fig. 4.2) in order to perform safe and flexible vehicle navigation. The waypoints are obtained from an efficient and flexible methodology based on multi-criteria optimization using either grid map (cf. Subection 4.3.3.1) or expanding tree (cf. Subection 4.3.3.2).

In the both proposed Optimal Multi-criteria Waypoint Selection (OMWS) (i.e., based on Grid Map (GM) and Expanding Tree (ET)), waypoints are selected considering safe position on the road, feasibility of trajectories (smooth changes between the successive points and vehicle constraints) and localization uncertainties.

The waypoints assignment strategies (cf. Subsection 4.3.3.1 and 4.3.3.2) are formulated as an optimization problem and solved using dynamic programming [13] (cf. Formulation 1).

Formulation 1 (Optimization problem)**.** *For each discrete state $x_k \in X$ where X is a nonempty and finite state space. The objective is to obtain the sequence of states to reach the final state x_K while minimizing the following cost function:*

$$C(x_K) = \sum_{k=1}^{K} g(Pred_{x_k} \rightarrow x_k) + h(x_K) \tag{4.3}$$

where $Pred_{x_k}$ is the predecessor state of x_k. g is the immediate traveling cost function to go from $Pred_{x_k}$ to x_k. h is the future traveling cost function (heuristic) to go from the current state to the final state x_K. When the current state is the final state x_K then $h(x_K)$ is equal to zero. This function h contributes to improve the convergence of the suboptimal solutions towards the global optimal one [71].

4.3.3.1/ OMWS BASED ON GRID MAP (GM)

Before describing the proposed algorithm and the criterion to optimize, let us provide some useful definitions. A grid map corresponds to a limited area of the environment decomposed generally on square cells [192] (cf. Fig. 4.13 and 4.14). Each cell of the grid map can be an obstacle or a free space (cf. Section 4.3 for definition of C-space$_{obst}$ and C-space$_{free}$). The exterior limits (C-space$_{ext}$) of the C-space$_{free}$ area are defined by the user, even for open environment (cf. Fig. 4.2). For simplification, the dimension of the cells in the grid map is chosen according to the vehicle's dimension. Therefore, the vehicle is contained, at each sample time, in only one cell [192].

We consider the center of the cell (i, j) as its position. Each $cell_{ij}$ is defined by the following parameters:

- $\bar{w}_{ij} \in [0, 1]$ is related to the normalized distance $d_{ij\_To\_Obst}$ to the closest C-space$_{obst}$ or C-space$_{ext}$. $\bar{w}_{ij}$ is given by:

$$\bar{w}_{ij} = 1 - \frac{d_{ij\_To\_Obst}}{d_{max\_To\_Obst}} \tag{4.4}$$

  where $d_{max\_To\_Obst}$ is the maximum value among all $d_{cell\_To\_Obst}$ of all cells in the C-space$_{free}$. For instance, Fig. 4.14 shows different distances of cells localized at (a, b), (i, j) and (m, n) to the C-space$_{obst}$ or C-space$_{ext}$. $d_{max\_To\_Obst}$ is equal, in this example, to the maximum distance $d_{mn\_To\_Obst}$.

- *State* is the cell state, which has three possible values, $Init$ (Initialization), $Open$ (when it is in the expansion queue) and $Close$ (when it has already been expanded).

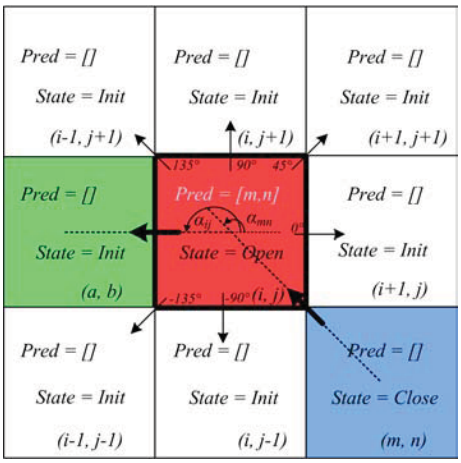


Figure 4.13: A group of cells of the global grid map, the current $cell_{ij}$ (red), its predecessor cell (blue) and its probable successive cell (green). *State* is the cell state and $Pred$ is the predecessor of the cell.

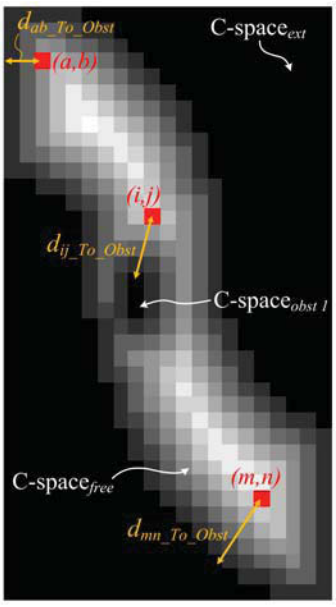


Figure 4.14: Representation in gray scale w.r.t. the distance to the closest C-space$_{obst}$ (the whitest area represents the safest area).

- Its set of neighbors are defined by:

$$S_N(cell_{ij}) = \{(i \pm \Delta i_N, j \pm \Delta j_N) | (\Delta i_N, \Delta j_N) \neq (0,0)\} \tag{4.5}$$

  where $\Delta i_N, \Delta j_N = 1 \ldots, N_h$. N_h is the neighborhood value (cf. Fig. 4.15). Fig. 4.15(a) shows the case where $N_h = 1$ (which implies 8 neighbor cells). Fig. 4.15(b) shows a larger neighbor cells when $N_h = 2$ (24 neighbors).

- $Pred_{ij}$ is the neighbor cell of ij which minimizes the total cost $C(cell_{ij})$ eq. (4.3) (cf. Fig. 4.13).

- $g(Pred_{ij} \rightarrow ij)$ is the traveling cost from the predecessor cell until the current $cell_{ij}$.

- $h(ij)$ is the heuristic traveling cost from the current $cell_{ij}$ to the final cell. As conventional, this cost depends on the euclidean distance from the $cell_{ij}$ to the final cell.

The traveling cost function $g(Pred_{ij} \rightarrow ij) = g(mn \rightarrow ij)$, from mn to ij, is normalized ($g \in [0,1]$). This function is also designed to take into account the variation of cell orientation (cf. Fig. 4.13). This variation allows to obtain an optimal path consisting of minimal number of straight lines. Therefore, a lowest possible number of waypoints in the safe area can be extracted from this optimal path. The cost function $g(mn \rightarrow ij)$ is given by:

$$g(mn \rightarrow ij) = k_g \bar{w}_{ij} + (1 - k_g) \frac{|\alpha_{ij} - \alpha_{mn}|}{2\pi} \tag{4.6}$$

where the first term of (4.6) is related to the safety of the obtained solution, and the real constant $k_g \in [0,1]$ is used to increase or to decrease the significance of this term. The second term of (4.6) is related to the smoothness of the obtained solution, i.e, the path has a limited and minimum orientation change. The cell orientations $\alpha_{mn},\ \alpha_{ij} \in]-\pi, \pi]$ are computed using the position of the current cell (i,j), its predecessor (m,n) and its probable successor (a,b) (cf. Fig. 4.13). They are computed as:

$$\alpha_{ij} = \arctan\left(\frac{a-i}{b-j}\right) \tag{4.7}$$

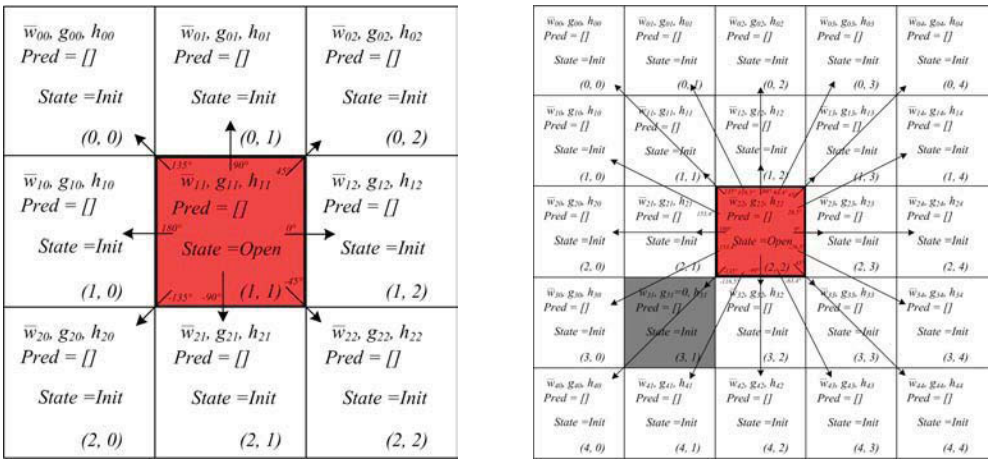
$$\alpha_{mn} = \arctan\left(\frac{i-m}{j-n}\right) \tag{4.8}$$

The heuristic traveling cost $h(ij) \in [0,1]$ (refers to eq. (4.3)) is designed in function of the euclidean distance d_{ij} from the $cell_{ij}$ to the final cell. This cost function is also used for OMWS-ET (cf. Subsection 4.3.3.2). The cost function $h(ij)$ is given by:

$$h(ij) = k_h \left(1 - e^{-d_{ij}/k_e}\right) \tag{4.9}$$

where $k_h \in [0,1]$ allows to tune the significance of $h(ij)$ in the total cost function (cf. eq. (4.3)). The exponential function was chosen because it can be formulated to give values between 0 and 1 for positive values of d_{ij}. The constant $k_e \in \mathbb{R}^+$ is used to scale the value of d_{ij} according to the environmental dimensions. The value of $h(ij)$ (cf. eq. (4.9)) decreases while the next selected cell goes closer to the final cell.

Algorithm 2 shows in pseudocode, the first proposed method to obtain the set of waypoints in a structured environment. It starts from the final vehicle's position (initial cell). The algorithm selects the cells that have the lower total cost $C(ij)$ (4.3) until to reach



(a) $N_h = 1$. (b) $N_h = 2$.

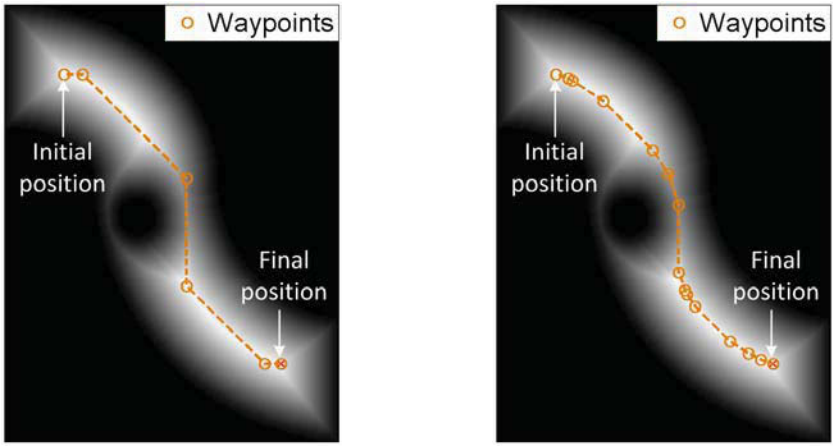
Figure 4.15: Different neighborhood values N_h of the current cell (red). A gray cell represents a cell where the movements are forbidden.

the final cell. A discrete path r is obtained while tracking the predecessor cell of each selected cell which minimizes the total cost starting in p_i until p_f. Finally, the proposed Algorithm 1 is applied to the discrete path r to obtain the set of waypoints S_p characterized by $(x_{p_i}, y_{p_i}, \theta_{p_i}, v_{p_i})$. Basically, this algorithm keeps only the sorted waypoints (pose, velocity and predecessor) which have an orientation changes w.r.t. its predecessors.

Algorithm 2 Waypoint selection based on a grid map

Require: Initial position p_i, final position p_f and a $Gridmap$
Ensure: Set of waypoints S_p

Init $State_{ij} = INIT$, $g_{ij} = 0$ and $Pred_{ij} = \oslash$, $\forall\ ij \in Gridmap$
Init $cell_{ij} = p_f$ and the set of neighbors $S_N(cell_{ij})$
while $cell_{ij} \neq p_i$ **do** ▷ Until to reach the initial position
    Set $State_{ij} = CLOSE$
    for $cell_N \in S_N(cell_{ij})$ **do**
        if $\bar{w}_{cell_N} \neq 0$ **then** ▷ Only cells in the free space
            Obtain $Pred_{cell_{ij}} = Predecessor(cell_{ij})$ ▷ When $cell_{ij} \neq p_f$
            Compute α_{ij} (4.7) and α_{mn} (4.8) ▷ These values are 0 when $cell_{ij} = p_f$
            Compute the total cost $C(cell_N)$ (4.3)
            if $State_{cell_N} == INIT$ **then**
                Add $cell_N$ to the queue Q and $Pred_{cell_N} = cell_{ij}$
                Set $State_{cell_N} = OPEN$
            else if $State_{cell_N} == OPEN$ **then** ▷ $cell_N$ is in Q with certain $C(cell_N)$ and $Pred_{cell_N}$
                Update $cell_N$ in the queue Q with the lower total cost $C(cell_N)$
                $Pred_{cell_N}$ is the cell which generates the lower $C(cell_N)$
            end if
        end if
    end for
    Sort the queue Q in ascending order of total cost C
    Get the first value of queue $cell_{ij} = Q(first)$ and remove it from Q
end while
Obtain the discrete path (straight lines) linking $r = p_i, Pred_{p_i}, Pred\{Predp_i\}, \ldots, p_f$.
Apply Algorithm 1 using r to obtain the set of waypoint S_p.



(a) $N_h = 1$. (b) $N_h = 3$.

Figure 4.16: Different sets of waypoints (red points) for different number of possible neighbor cells.

The obtained path is defined by straight lines connecting each two consecutive waypoints (blue dashed lines in Fig. 4.16) which belong to the set of obtained waypoints (cf. Algorithm 2). The smoothness of the path depends on the number of possible neighbors of the expanded cell defined by N_h (cf. Fig. 4.16). The drawback of using a large number of neighbors is obviously the increase of processing time. When $N_h \geq 1$, it is mandatory to check if the current neighbor is blocked by some other forbidden neighbor (cf. Fig. 4.15(b)). For an off-line planning, $N_h > 1$ can always be used to obtain a coherent and optimal solution regardless of time consumption.

4.3.3.2/ OMWS BASED ON EXPANDING TREE (ET)

This subsection presents in details the proposed algorithm for optimal planning of the vehicle's path, using an appropriate expanding tree. The formulation of this expanding tree integrates the kinematic model of the vehicle as well as the proposed control law definition (cf. Section 3.5) and the vehicle's localization uncertainties.

Before describing the proposed method and the criterion to be optimized, let us present the definition of expanding tree. The expanding tree is composed by nodes and edges which have the following properties:

- Each node q_j is defined by its pose $(x_{q_j}, y_{q_j}, \theta_{q_j})^T$, one predecessor node q_i (except for the initial node) and a traveling cost values $g(.)$ and $h(.)$ (cf. eq. (4.3)).
- Each edge ξ_{ij} corresponds to the link between q_i to q_j nodes.
- $g(q_i \rightarrow q_j) = g(\xi_{ij})$ is the traveling cost from q_i to q_j.
- $h(q_j)$ is the heuristic traveling cost from the current node q_j to the final node (final vehicle pose). It was defined by eq. (4.9) (cf. Subsection 4.3.3.1).

The traveling cost $g(\xi_{ij}) \in [0, 1]$ is designed to obtain an appropriate balance among safe, smooth, feasible and fast vehicle's trajectory. It is defined as:

$$g(\xi_{ij}) = k_1 \bar{w}_j + k_2 \Delta\bar{v}_{ij} + k_3 \Delta\bar{\gamma}_{ij} + k_4 \Delta\bar{e}_{l_{ij}} \tag{4.10}$$

where $k_1,\ k_2,\ k_3$ and $k_4 \in \mathbb{R}^+$ are constants which are defined by the designer to give the right balance (according to context of navigation, e.g., give more priority to the safety with regard to the smoothness) of each term of the criterion (cf. eq. (4.10)). To normalize the traveling cost, $k_i | i = 1, \ldots, 4$ must satisfy:

$$k_1 + k_2 + k_3 + k_4 = 1 \tag{4.11}$$

The normalization of the individual criterion given in (4.10) allows to simplify the choice of k_i to select the priority of a term w.r.t. the others according to the navigation context. Different set of values $k_1,\ k_2,\ k_3$ and k_4 will be considered for different scenarios in Subsection 4.3.4.

The first term of the cost function eq. (4.10) is related to the navigational safety (4.4). The second and third terms are respectively related to the speed (4.12) and smoothness (4.15) of the trajectory. The fourth term is related to feasibility of the vehicle's trajectory while considering localization uncertainties, i.e., the risk to collide with an obstacle while considering inaccuracies in the UGV's pose (a detailed explanation of this term is given later in this subsection). This last term allows to consider the kinematic model of the UGV and the control law. The details of each term are given in the following:

- The term $\bar{w}_j \in [0, 1]$ is related to the distance from the node q_j to the closest C-space$_{obst}$ or C-space$_{ext}$. It is given by eq. (4.4) (cf. Subsection 4.3.3.1).

- The term $\Delta\bar{v}_{ij} \in [0, 1]$ is related to the velocity v_{ij} from q_i to q_j. It is given by:

$$\Delta\bar{v}_{ij} = 1 - \frac{v_{ij}}{v_{max}} \tag{4.12}$$

  where v_{max} is the maximum velocity of the vehicle. v_{ij} is estimated as a function of the curvature of the trajectory. The maximum velocity occurs when the curvature is zero (straight line) and the minimum velocity $v_{min} \neq 0$ occurs when the curvature is bigger than the value corresponding to γ_{max} (cf. Fig. 3.5). This consideration allows the vehicle to maneuver without risk of collisions while enhancing the passenger comfort since the centripetal forces are limited [254]. The minimum and maximum values of velocity and steering angle are defined by the designer according to vehicle's characteristics. The curvature is estimated using the orientation of the current node and its predecessor. Therefore, v_{ij} is computed as:

$$v_{ij} = v_{max} - \Delta\bar{\theta}_{ij}(v_{max} - v_{min}) \tag{4.13}$$

  where $\Delta\bar{\theta}_{ij} \in [0, 1]$ is the normalized estimated curvature related to the variation of orientation between the current node q_j and its predecessor q_i. It is defined as:

$$\Delta\bar{\theta}_{ij} = \frac{|\theta_j - \theta_i|}{\Delta\theta_{max}} \tag{4.14}$$

  where $\Delta\theta_{max}$ is the maximum variation between a probable orientation of the current node w.r.t the orientation of its predecessor. This value is defined according to the steering vehicle's capability. θ_j and θ_i are computed using the node positions and eq. (4.7) and (4.8) (cf. Subsection 4.3.3.1).

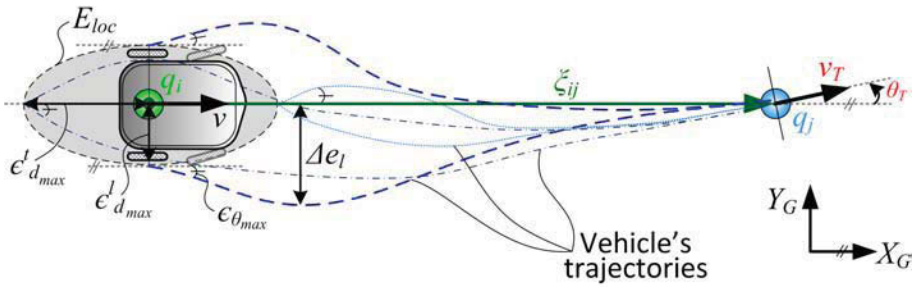


Figure 4.17: Vehicle's trajectories which starts from extreme configurations ($\pm\epsilon^l_{d_{max}}$, $\pm\epsilon^t_{d_{max}}$ and $\pm\epsilon_{\theta_{max}}$) in the ellipse of localization uncertainties E_{loc}.Δe_l is the maximum lateral deviation of all vehicle's trajectories.

- The term $\Delta\bar{\gamma}_{ij} \in [0,1]$ is related to the variation of steering angle along the vehicle's trajectory from q_i to q_j. For instance, Fig. 4.17 shows a vehicle's trajectory between two nodes. It is given by:

$$\Delta\bar{\gamma}_{ij} = \frac{\sum_{q_i}^{q_j} |\Delta\gamma_{ij}|}{n_{q_{ij}}\gamma_{max}} \tag{4.15}$$

  where $n_{q_{ij}}$ is the considered point number of the vehicle trajectory between q_i and q_j, and γ_{max} is the maximum steering angle of the vehicle. This term $\Delta\bar{\gamma}_{ij}$ (4.15) computes the sum of the $\Delta\gamma_{ij}$ to obtain the total variation of the steering angle along the vehicle's trajectory. $\Delta\bar{\gamma}_{ij}$ uses the kinematic model and the control law to estimate the vehicle's trajectory.

- The term $\Delta\bar{e}_{l_{ij}} \in [0,1]$ is the normalized maximum deviation of vehicle's trajectory w.r.t. the straight line that joins the positions (x_q, y_q) of q_i and q_j (cf. Fig. 4.17). It is computed as:

$$\Delta\bar{e}_{l_ij} = \frac{|\Delta e_{l_{ij}}|}{max\{\Delta e_l\}} \tag{4.16}$$

  where $max\{\Delta e_l\}$ is the maximum deviation of all trajectories from the node q_i to the node q_j while considering the position and orientation uncertainties (ϵ_d and ϵ_θ respectively given in Fig. 4.17). This term $\Delta\bar{e}_{l_ij}$ allows to estimate the collision risk using the vehicle's trajectory that takes into account the kinematic model, the control law and localization uncertainties (position and orientation). Fig. 4.17 shows an illustration where the UGV has an ellipse of localization uncertainties with axes ϵ^l_d and ϵ^t_d. The UGV's trajectories start at $\pm\epsilon^l_d$ in lateral distance (longitudinal distance is set to 0), and $\pm\epsilon^t_d$ in longitudinal distance (lateral distance is set to 0) from the UGV's position with a $\pm\epsilon_\theta$ from the presumed vehicle's orientation, i.e., we consider all extreme configurations to obtain, according to these maximum error configurations, the maximum lateral deviation (Δe_l). The trajectories are obtained using the kinematic model and the proposed control law in an offline simulated procedure.

Algorithm 3 shows in pseudocode, the proposed method which uses expanding tree to obtain the optimal waypoints configurations w.r.t. the optimized multi-criteria function eq. (4.10). Fig. 4.18 shows the first steps of the algorithm in which, for instance, the branch numbers of each node is $n_t = 3$. Each branch orientation w.r.t. the vehicle orientation is given by:

$$\alpha = \pm i\Delta\alpha, \quad i = \begin{cases} 0, 1, \ldots, (n_t - 1)/2; & \text{if } n_t \text{ is odd} \\ 1, 2, \ldots, n_t/2; & \text{if } n_t \text{ is even} \end{cases} \tag{4.17}$$

Algorithm 3 Waypoint selection based on expanding tree

Require: Initial pose p_i, final pose p_f, branch number n_t, edge distance ξ, branch orientation $\Delta\alpha$, tolerable error distance ϵ and C-space$_{free}$
Ensure: Set of waypoints S_p
1: Init the initial node $q_0 = p_i$, $g_0 = 0$ and $Pred_{q_0} = \oslash$
2: Init the current node to expand $q_i = q_0$
3: Init $Tree(q_i)$ =Expansion_Tree (Procedure 4) with $\alpha = 0$ ▷ Initial expansion
4: Set the new node to expand $q_i = r_t$ where $r_t \in Tree(q_i)$
5: Set $Pred_{r_t} = q_i$ and compute the total cost $C(r_t)$ eq. (4.3)
6: **while** $|q_i - p_f| < \epsilon$ **do**
7: Compute the $Tree(q_i)$ = Expansion_Tree ▷ Procedure 4 with a set of $\alpha = \pm i\Delta\alpha$
8: **for** $r_t \in Tree(q_i)$ **do**
9: **if** $r_t \in$ C-space$_{free}$ **then**
10: Compute the total cost $C(r_t)$ eq. (4.3)
11: $Pred_{r_t} = q_i$
12: Add r_t to the queue Q
13: **end if**
14: **end for**
15: Sort the queue Q in ascending order of total cost C
16: Get the first value of queue $q_i = Q(first)$ and remove it from Q
17: **end while**
18: Obtain the discrete path (straight lines) linking $r = p_i, \ldots, Pred\{Predp_f\}, Pred_{p_f}, p_f$.
19: Apply Algorithm 1 using r to obtain the set of waypoint S_p.

where $\Delta\alpha$ is a constant angle defined according to the vehicle's characteristics.

The edge distance ξ is the Euclidean distance between two successive nodes and it depends on the environment dimensions, e.g., if the environment has a narrow passage then ξ must cope with this dimension. It is considered that the edge orientation is the vehicle's orientation at the current node position (cf. Fig. 4.18). Thus, at beginning the first expansion of q_0 is given with $\alpha = 0$ because the vehicle starts at initial fixed pose (cf. line $3-5$ of Algorithm 3). This initial expansion is made to respect the kinematic constraints where the vehicle's rotation requires a displacement (linear velocity $\neq 0$) of it. Therefore, the successive node q_1 has different possible orientation and so on (cf. Fig. 4.18). The algorithm selects the node which has the lower total cost $C(q_j)$ (4.3). When two or more nodes have the same cost, the algorithm selects the last saved node. Fig. 4.18 shows the successive steps, the node q_4 was selected from the expansion of q_1 $\{q_2, q_3, q_4\}$ since it has the lower total cost value. The selection of the node with lower total cost (cf. Algorithm 3, line $15-16$) allows to avoid the deadlock areas because the successive branches from the nodes in this deadlock area will be in C-space$_{obst}$ (cf. Fig. 4.19(a)).

Procedure 4 Expansion_Tree

Require: Current node q_i, set of α $S(\alpha)$, edge distance ξ
Ensure: Nodes of $Tree(q_i)$
1: Init $Tree(q_i) = \oslash$
2: **for** $\alpha_t \in S(\alpha)$ **do**
3: Compute orientation $\theta_{r_t} = \theta_{q_i} + \alpha_t$
4: Compute pose $r_t = q_i + [\xi\cos(\theta_{r_t}), \xi\sin(\theta_{r_t}), \alpha_t]^T$
5: Add r_t to $Tree(q_i)$
6: **end for**

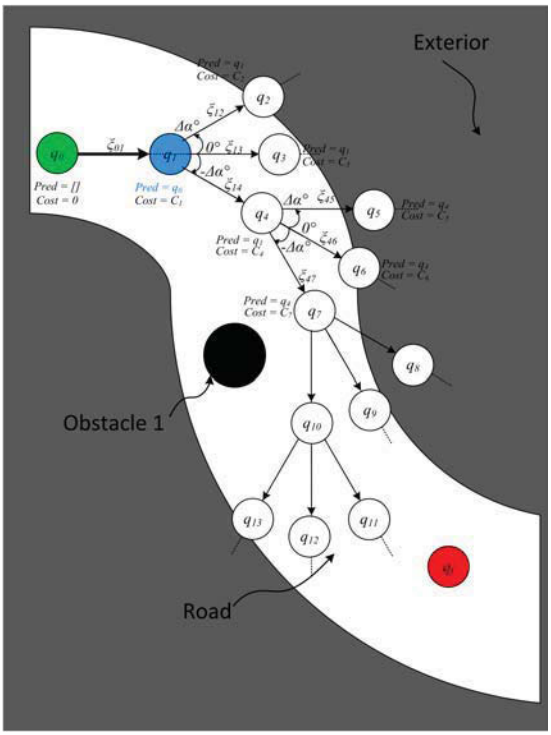
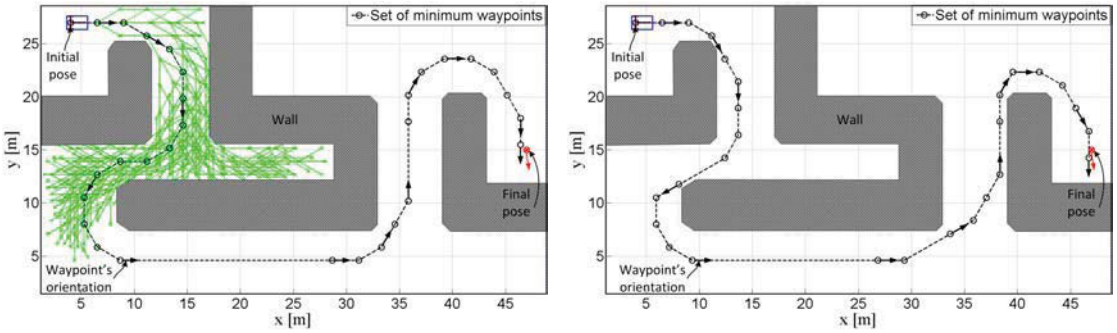


Figure 4.18: Expanding tree method to obtain the appropriate set of waypoints.

A discrete path r is obtained while tracking the predecessor cell of each selected cell which minimizes the total cost starting in p_f until p_i. Finally, the proposed Algorithm 1 is applied to the discrete path r to obtain the set of waypoints S_p as detailed in Subsection 4.3.3.1. The smoothness of the UGV's trajectory depends on the number of branches of each tree n_t, the maximum branch orientation $\alpha_{max} = n_t \Delta\alpha/2$ and the edge distance ξ (cf. Subection 4.3.4). The drawback of using a large number of n_t is the increase of the processing time required to obtain the set of waypoints. The edge distance ξ is set to detect obstacles between the successive nodes.

This method uses deterministic selection of expanding tree to obtain the optimal solution with lowest total cost. Nevertheless, a feasible solution can be obtained using a probabilistic selection of expanding tree to decrease the processing time (cf. Subsection 4.3.4), i.e., the branch orientation α and edge distance ξ are randomly selected in a fixed interval [221]. Simulations in Subsection 4.3.4 show the case where these parameters are randomly chosen.

As described above, the traveling cost eq. (4.10) depends on four parameters $(k_i | i = 1, \ldots, 4$, which satisfy eq. (4.11)) related respectively to the safety, velocity, less steering and taking into account localization uncertainties. The values of these parameters are fixed according to the desired navigation and environment conditions. A pragmatical procedure to set these parameters consists in first identifying the main desired vehicle behavior and setting its parameter k_i with a value greater than 0.5 (cf. Fig. 4.19). The other parameters will be tuned according to the designer's secondary priorities. Fig. 4.19 shows the set of waypoints when only the term with highest priority is considered in the traveling cost function. For instance, the priority is given to the safest and the shortest paths represented respectively in Fig. 4.19(a) and 4.19(b). More examples of different tuned parameters will be shown in Section 4.3.4.



(a) Safest path: $k_1 = 1.0$, $k_2 = k_3 = k_4 = 0.0$ and $k_h = 0.1$

(b) Shortest path: $k_2 = 1.0$, $k_1 = k_3 = k_4 = 0.0$ and $k_h = 0.1$

Figure 4.19: Set of waypoints for different parameters values k_i of the traveling cost.

4.3.4/ SIMULATION VALIDATION

This subsection shows optimal sets of waypoints, obtained according to the environment characteristics and/or the task to achieve. In what follows, it will be shown: a comparison between grid map and expanding tree algorithms (cf. subsection 4.3.4.1) and a comparison between the proposed OMWS-ET and a variation of RRT (cf. subsection 4.3.4.2). Several other simulations to demonstrate the efficiency of the OMWS-ET method for planning of UGV's navigation in a structured environment are also exhibited such as trajectory generation (cf. Subsection 4.3.4.3); a comparison between deterministic and probabilistic waypoints selection (cf. Subsection 4.3.4.4), extension to multi-robot formation (cf. Subsection 4.3.4.5) and finally, Subsection 4.3.4.6 shows the flexibility of our proposal for local replanning of the waypoints configurations when unexpected obstacles are detected.

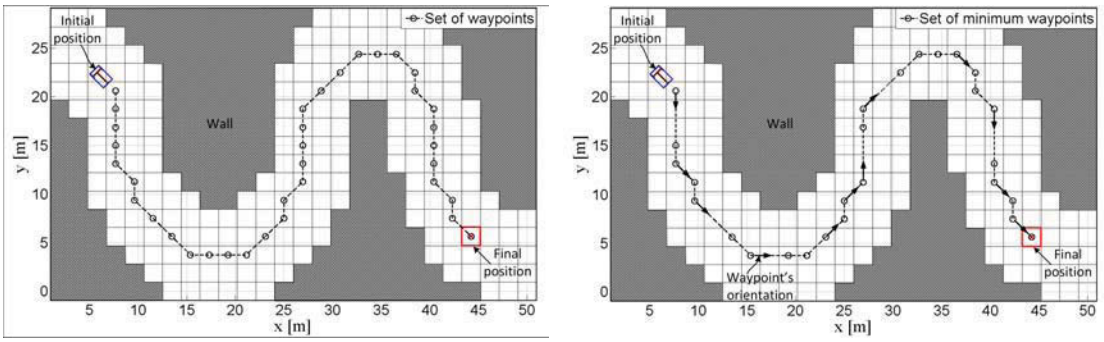
For these simulations, the physical parameters, constraints and controller parameters **K** of the UGV were described in Subsection 4.2.3.

4.3.4.1/ GRID MAP VERSUS EXPANDING TREE

These simulations show two set of waypoints obtained by two proposed methods based on grid map and expanding tree (cf. Algorithms 2 and 3 respectively). The maximum number of iteration is $n_I = 5000$ to stop both algorithms, OMWS-GM (Algorithm 2) and OMWS-ET (Algorithm 3), when none solution can be obtained. Fig. 4.20(a) and Fig. 4.21(a) show the obtained path according to Algorithm 2 and 3 before to apply Algorithm 1. The minimum set of waypoints are given afterward in Fig. 4.20(b) and Fig. 4.21(b).

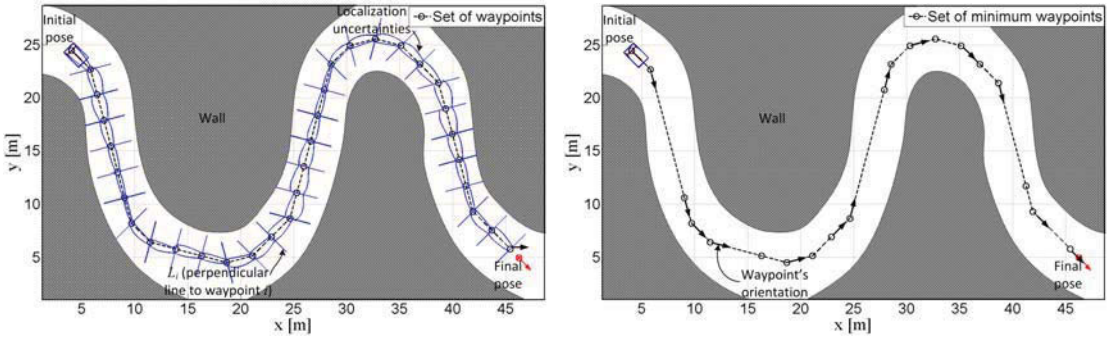
For the grid map case, the cell has the UGV's dimension $(2\ m)$ and its neighborhood is $N_h = 1$. The constant value of k_g is 0.6 eq. (4.6) and k_h is 0.1 eq. (4.9). The minimum set of waypoints has $n_p = 27$ elements. An additional constraint is added for OMWS-GM (before line 9 of the Algorithm 2), the angle variation (second term of eq. (4.6) must be less than a threshold $\Delta\alpha_{th}$). This constraint enables the processing time of the algorithm to be reduced since it considers only the cells with an orientation change, w.r.t the last cell orientation, less than $\Delta\alpha_{th}$ (cf. Fig. 4.13).

For Expanding Tree case, the branch number n_t is 5, the edge distance ξ is $2.5\ m$ and $\Delta\alpha$ is $15°$. The safety gain k_1 (cf. eq. (4.10)) is considered as the highest priority in this simulation. The constant values of $k_i | i = 1, \ldots, 4$ (4.10) are $k_1 = 0.6$, $k_2 = 0.2$, $k_3 = 0.1$,



(a) (b)

Figure 4.20: a) Obtained waypoints using Algorithm 2 based on grid map before applying Algorithm 1 and b) Minimum set of waypoints.



(a) (b)

Figure 4.21: a) Obtained waypoints using Algorithm 3 based on expanding tree before applying Algorithm 1 and b) Minimum set of waypoints.

$k_4 = 0.1$ and $k_h = 0.1$ eq. (4.9).

Table 4.1 shows different performance criteria to compare the set of waypoints where: n_p is the number of waypoints, $length$ is the sum of distance between two successive waypoints, d_{border} is the sum of minimum distance to the road boundaries. Therefore, the method based on OMWS-ET is more safe, accurate and efficient than the one based on OMWS-GM, mainly when the criterion to optimize is related to the UGV's model (velocity and steering angle).

It is noted that the set of waypoint obtained by OMWS-ET is smaller (19 elements, cf. Table 4.1) than the method OMWS-GM which does not consider the orientation neither the UGV's model. To avoid a large growing of the tree branch of OMWS-ET, a position and orientation comparison between nodes can be added at line 12 of the Algorithm 3. If two nodes from different branches have the same position and orientation then the node with lowest total cost value (cf. eq. (4.3)) is stored and the other node is removed.

	n_p	$length$ [m]	d_{border} [m]
OMWS-GM	27	77.22	52.1268
OMWS-ET	19	77.50	56.7217

Table 4.1: Comparison between the OMWS-GM and OMWS-ET.

4.3.4.2/ OMWS-ET VERSUS RRT*

To highlight the advantages and flexibility of the proposed OMWS-ET, a comparison with the popular RRT* algorithm [301] is presented in this subsection. The RRT* is based on the RRT (Rapidly-exploring Random Tree) already described in Section 4.3 with an addition of the rewiring function which allows to reconnect the nodes to ensure that the edges have the path with minimum total cost (cf. Appendix B.2). RRT* provides thus an optimal solution with minimal computational and memory requirements [301]. Moreover, RRT* is a sampling-based algorithm and the optimal solution depends on the number of iterations of the RRT* algorithm, i.e., more is the number of iterations (more samples in the C-space$_{free}$) closer is the obtained solution to the effective optimal global solution. Therefore, to compare the solutions obtained by the OMWS-ET with those obtained by the RRT* some little modifications in Algorithm 3 were made. The line 6 of Algorithm 3 was changed by a *for* loop from 0 to the maximum iteration number n_I and the selection of the final pose at each iteration is obtained by the sampling in C-space$_{free}$ (q_{random}) as the RRT* Algorithm [301]. It is to be noted that q_{random} corresponds to a random sample (position) from a uniform distribution in the C-space$_{free}$.

To compare OMWS-ET and RRT* algorithms, the safest obtained solution (which maximizes the distance to the border) is used as criterion. Therefore, the parameters of the cost function of OMWS-ET eq. (4.10) are fixed to: $k_1 = 1.0$, $k_2 = k_3 = k_4 = 0.0$ and $k_h = 0.1$. In addition, the other parameters are fixed such as: the branch number $n_t = 5$, the edge distance $\xi = 2.5\ m$ and $\Delta\alpha = 15°$.

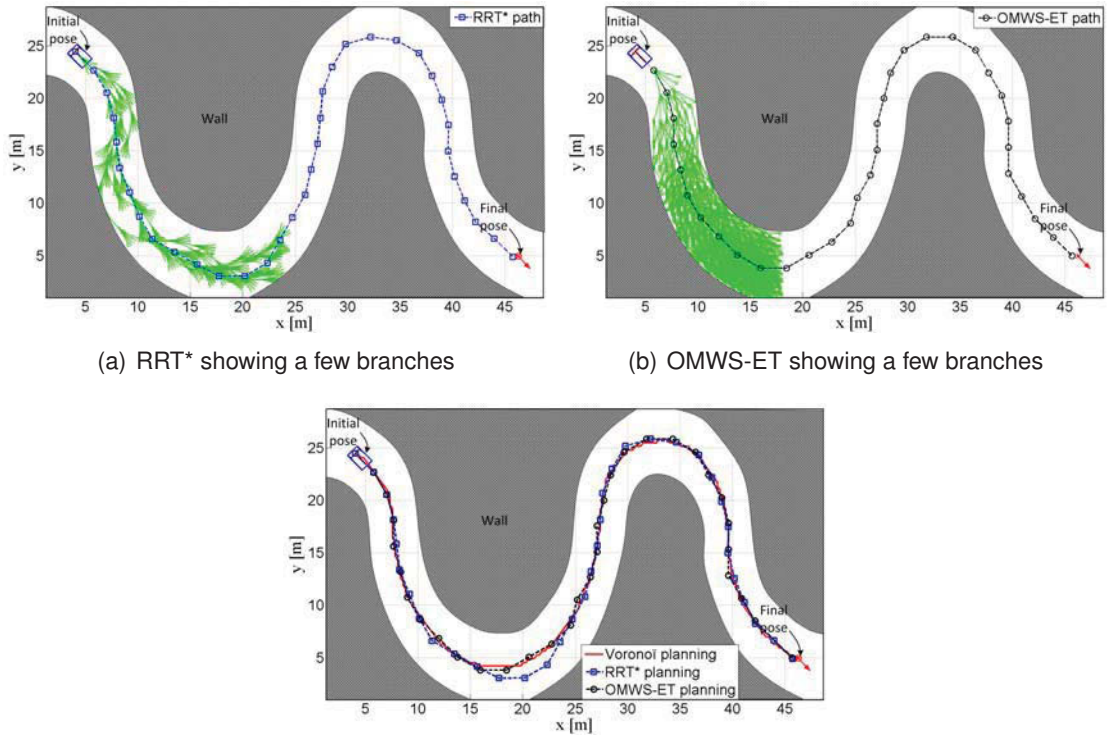
The RRT* algorithm described in [301] (cf. Appendix B.2) was also modified to obtain a cost function according to the safety $\bar{w}_i$ (distance to the border) instead of an Euclidean distance between nodes. The kinematic model (cf. Fig. 3.5) with constant linear velocity and steering angles ($v = 1.0\ m/s$ and $\gamma = -15, -7.5, 0, 7.5, 15°$) respectively, during $t_{exp} = 2.5\ s$ was used to produce the new nodes of the RRT*. The maximum number of iteration for both algorithm is fixed to $n_I = 5000$.

Fig. 4.22 shows the obtained path solutions according to RRT*, OMWS-ET and also Voronoï [46] algorithms. The obtained Voronoï path (cf. Fig. 4.22(c)) is represented since it is the best reference w.r.t. the adopted comparison criterion (safety). Indeed, Voronoï path permits always to obtain the safest possible path [46].

It can be noted that two obtained path using RRT* and OMWS-ET are generally enough far from the way borders (cf. Fig. 4.22(a) and 4.22(b)). Nevertheless, significant differences are observed in the obtained final results (cf. Fig. 4.22(c)). In fact, the obtained set of waypoint using RRT* are closer to the border which is due to the fact that RRT* expands its branches while adopting constant commands (v, γ, t_{exp}). These constant commands generate the next nodes with only a single possible orientation (for each node). Contrary to that, in the proposed OMWS-ET, each new obtained node q_j has different possible orientations and velocities, thus, for the same position, more possible UGV's states (different orientations and velocity set-points) are taking into account in the optimization process.

Table 4.2 shows, as in the last subsection, different performance criteria to compare the obtained path. It is shown that the obtained path based on OMWS-ET is closer than the RRT* to the optimal obtained solution using Voronoï method. This result validates that the proposed OMWS-ET is more efficient than the RRT*, in the sens that OMWS-ET explores more possibilities in the state space of the UGV and environment.

Moreover, it is important to mention that the proposed OMWS-ET algorithm is related to



(a) RRT* showing a few branches

(b) OMWS-ET showing a few branches

(c) Comparison with Voronoï

Figure 4.22: Three obtained path according to Voronoï, RRT* and OMWS-ET.

the adopted navigation strategy (cf. Section 4.2), which uses set-points based on suitable static/dynamic waypoints instead of trajectory tracking methods. OMWS-ET method takes into account the UGV's kinematics constraints and localization uncertainties as well as the proposed control law (cf. subsection 4.3.3.2). RRT* method is more suitable for navigation strategies based on trajectory following [301].

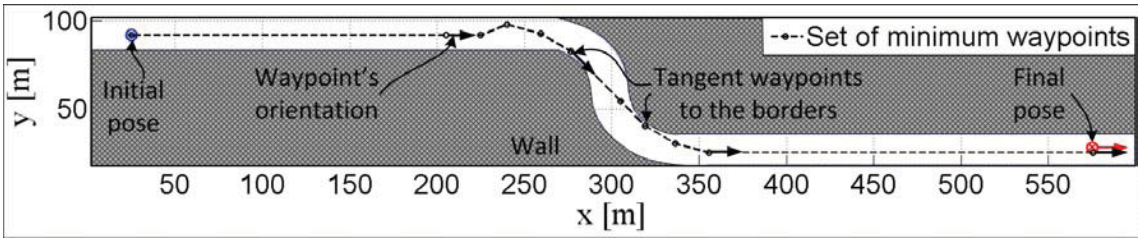
	$length$[m]	d_{border}[m]
Voronoï	86.00	69.2931
RRT*	83.42	62.1736
OMWS-ET	82.50	65.5926

Table 4.2: Comparison between Voronoï, RRT* and OMWS-ET.

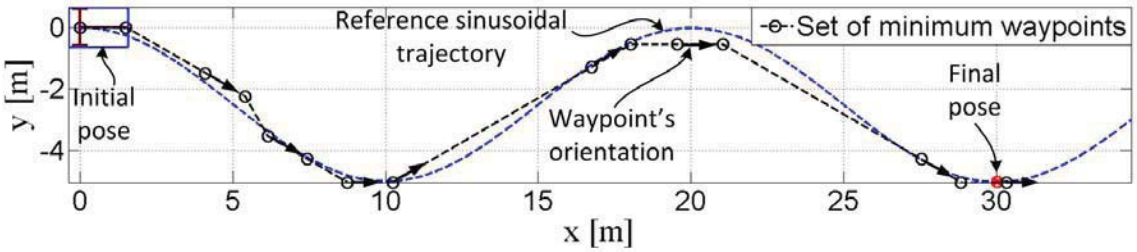
4.3.4.3/ SPECIFIC SCENARIO CASES

This subsection shows the minimum set of obtained waypoints for different scenarios using the method based on expanding tree (Algorithm 3). Fig. 4.23(a) shows the set of waypoints while considering the edge distance $\xi = 10\ m$ with an objective to obtain the fastest trajectory from the initial to the final positions, while not colliding with the road limits. The constant values are $k_1 = 0.1$, $k_2 = 0.7$, $k_3 = 0.1$, $k_4 = 0.1$ and $k_h = 0.1$. The minimum set of waypoints allows the vehicle to generate a minimum time trajectory as in [337]. This trajectory has a segment close to the route boundaries (tangent to the borders) which allows to navigate applying the maximum velocity.

Fig. 4.23(b) shows the use of the proposed OMWS-ET for the specific case where a



(a) Minimum set of waypoints for fastest trajectory.



(b) Minimum set of waypoints in a reference path.

Figure 4.23: Different scenarios for OMWS-ET.

reference path already exists for the UGV's navigation. In this case, the set of waypoints will be chosen as close as possible to the considered path (it depends on the chosen values of ξ and $\Delta\alpha$ in Algorithm 3). The set of waypoints obtained using OMWS-ET allows thus more flexible and safe navigation of the UGV between the waypoints (cf. criterion to optimize in eq. (4.10)). The edge distance ξ is set to $1\ m$ and $\Delta\alpha$ is 10° . The minimum set of waypoints are obtained while considering the term $\bar{w}_j$ (4.10) as the normalized minimum distance of the node q_j to the reference path. The constant values are set to $k_1 = 0.6$, $k_2 = 0.2$, $k_3 = 0.1$, $k_4 = 0.1$ and $k_h = 0.1$. The values of ξ and $\Delta\alpha$ can produce some waypoints outside the reference trajectory, e.g., if we decrease the values of ξ and $\Delta\alpha$ and increase the number of branches n_t then the waypoints will be on the reference trajectory. In Section 4.3.2, the waypoints are selected while considering only the points in the reference trajectory. It consists on analyzing the orientation variation of each points on the trajectory. In this case, the waypoints are selected in the environment to be close to the reference trajectory which allows to obtain less number of waypoints than the method used in Section 4.3.2 [4].

4.3.4.4/ DETERMINISTIC VERSUS PROBABILISTIC

This simulation shows the comparison between a deterministic and probabilistic expanding tree (i.e., where the values of ξ and α are probabilistically taken from an interval, instead of, to be fixed by the designer). Fig. 4.24 shows the minimum set of waypoints obtained using probabilistic expanding tree, where $\xi \in]0,\ 2.5]$ and $\alpha \in [-30^\circ, 30^\circ]$. The constant values are $k_1 = 0.6$, $k_2 = 0.2$, $k_3 = 0.1$, $k_4 = 0.1$ and $k_h = 0.1$. The processing time of the method with probabilistic expanding tree is less than the method with deterministic expanding tree. Nevertheless, the set of waypoints are not the optimal solution. The advantages of probabilistic selection of ξ and α is to reduce the convergence time and to obtain an online implementation [253, 291]. In future works, the choice of the variation of ξ and α will be oriented to improve the efficiency of the algorithm.

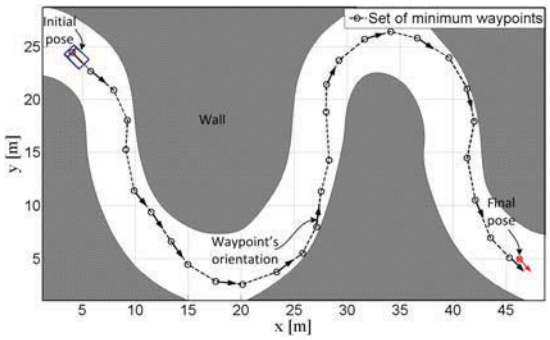


Figure 4.24: Set of waypoints using probabilistic expanding tree.

4.3.4.5/ OMWS-ET FOR MULTI-ROBOT FORMATION

The proposed method based on expanding tree (Algorithm 3) was extended to multi-robot formation (MRF) where the formation is defined only according to the leader's configuration [249] (cf. Fig. 4.25 and Subsection 5.2.1). As mentioned before, OMWS-ET algorithm takes into account the vehicle's model. To cope with this multi-robot task, it is sufficient to adapt the term $\Delta\bar{e}_{l_ij}$ (4.16) in order to consider all trajectories of the group of UGVs. Fig. 4.26 shows the minimum set of waypoints for a line formation (inter-vehicle distance $d_i = 6\ m$) with two vehicles. The constant values are the same as the last simulation. The set of waypoints for the leader UGV are close to the curve road boundaries because the formation needs enough space to turn while keeping the rigid formation shape. The follower (blue rectangle) is always inside of the road boundaries.

4.3.4.6/ LOCAL REPLANNING FOR UNEXPECTED OBSTACLES

The proposed OMWS-ET algorithm is adapted to local replanning when an unexpected static obstacle is detected in the environment. Fig. 4.27 shows the *Waypoint configuration selection* block of the proposed control architecture (cf. Fig. 4.11) to activate the replanning of the UGV's movements based on an initial set of waypoints already obtained using OMWS-ET.

The UGV starts the navigation through the successive waypoints (cf. Section 4.2) from the

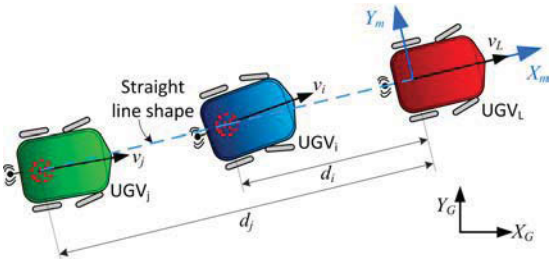


Figure 4.25: Multi-robot formation (straight line shape).

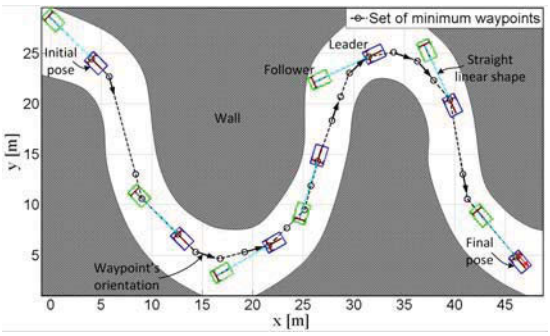


Figure 4.26: Minimum set of waypoints for multi-robot formation obtained by Algorithm 3 based on expanding tree.

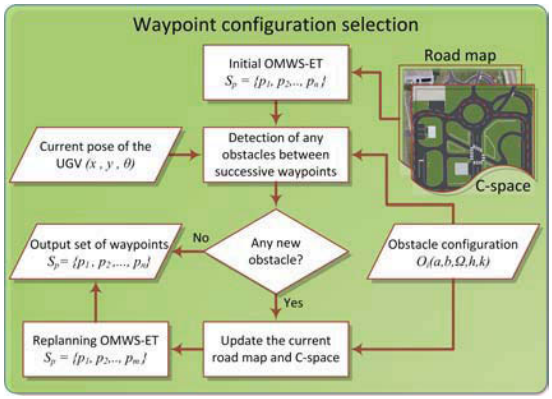
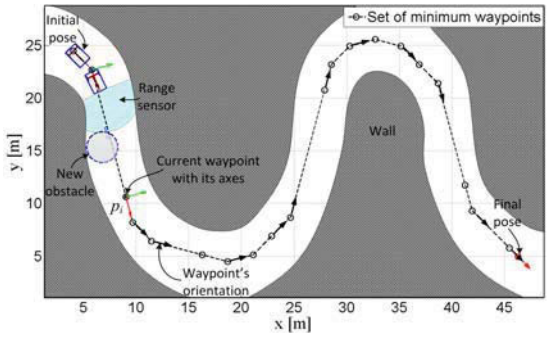
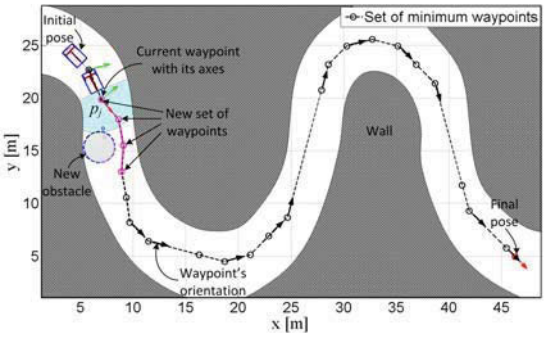


Figure 4.27: Flowchart of the *Waypoint configuration selection* block for local replanning.

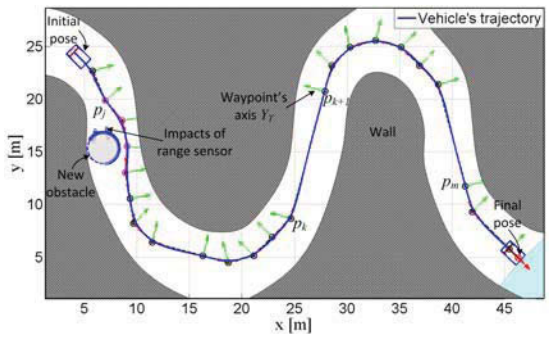
initial set of waypoints. These waypoints were already computed using the OMWS-ET in the known environment (cf. Subsection 4.3.3.2). The UGV uses a range sensor to detect any unforeseen obstacle (cf. Fig. 4.28(a)). A local replanning is activated when any new obstacle is detected. This replanning takes into account the current environment state, the current UGV's pose and the current waypoint to obtain a new local set of waypoints (cf. Fig. 4.28(b)). If the current waypoint is unreachable (due to the presence of the obstacle) then the final position is replaced by the next waypoint in the list and so on. If no solution is found then the vehicle will stop in its current pose. Figure 4.28(b) shows an example of the local replanning using the set of waypoints given in Subsection 4.3.4.1 (as initial set of waypoints) (cf. Fig. 4.21(b)). Finally, the vehicle moves through the new



(a) Unexpected obstacle is detected.



(b) Local replanning.



(c) Safe vehicle trajectory.

Figure 4.28: Local replanning for unexpected obstacle.

set of waypoints while guaranteeing a safe navigation (cf. Fig. 4.28(c)).

4.4/ EXPERIMENTAL VALIDATIONS

This section presents two experiments to demonstrate the efficiency of the proposed navigation strategy through successive waypoints (cf. Section 4.2) and the optimal selection of a set of waypoints based on expanding tree OMWS-ET (cf. Subsection 4.3.3.2) in structured environment. Subsection 4.4.1 focuses on the planning aspects using OMWS-ET and subsection 4.4.2 provides a complete experiment of the overall control architecture for vehicle's navigation in an urban environment containing hinder obstacles. These experiments were performed using a VIPALAB in PAVIN platform (Plate-forme d'Auvergne pour Véhicules INtelligents) (cf. Appendix G.1). The used UGV's parameters, constraints, sensors and controller parameters were described in Section 3.6. A metric map of PAVIN [357] is used by the proposed OMWS-ET (Algorithm 3). This map allows implementing the navigation through successive waypoints in a real vehicle and situations. The proposed OMWS-ET computes the set of geo-referenced waypoints with optimal configuration. Certain areas are restricted to guide the Algorithm 3 through PAVIN platform which has intersections and roundabout (cf. Fig. 4.29). In our case, these restricted areas were selected by the user, nevertheless the selection can be made by considering the topological map of the environment.

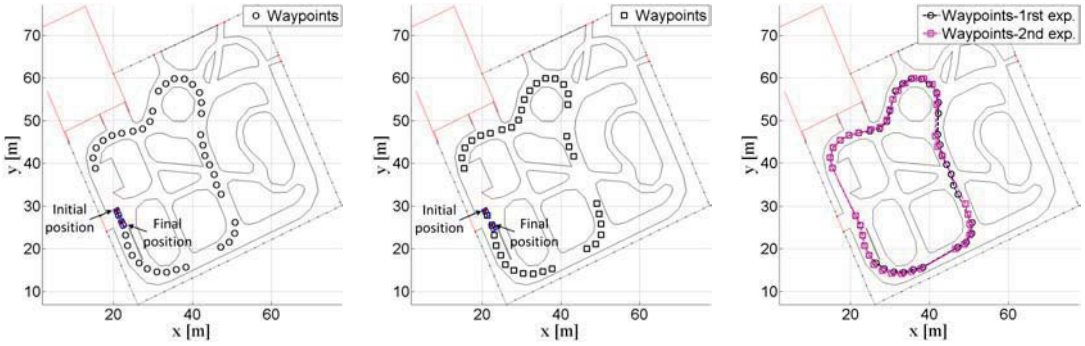
4.4.1/ WAYPOINT PLANNING USING OMWS-ET

This experiment makes a comparison between two cases of the proposed OMWS-ET: the first, corresponds to give more priority for the safety criteria in eq. (4.10 on page 96) and the second gives more priority for the minimum angle steering rate. The analysis of the obtained solutions will be given in what follows. Moreover, the actual vehicle's trajectories are compared for these different set of waypoints. This experiment can be found online[1].

Fig. 4.29 and 4.30 show respectively the minimum obtained set of waypoints and the corresponding actual vehicle's trajectories using the kinematic model of the UGV (simulation) and the VIPALAB (experiment). Fig. 4.29(a) shows the set of waypoints of the first experiment where the constant values of the cost function eq. (4.10) are $k_1 = 0.6$, $k_2 = 0.2$, $k_3 = 0.1$, $k_4 = 0.1$ and $k_h = 0.4$. The safety (k_1) has the highest priority in this experiment. Therefore, these waypoints guide the vehicle to be close to the middle of the road (cf. Fig. 4.30(a)). Fig. 4.29(b) shows the set of waypoints of the second experiment where the constant values are $k_1 = 0.3$, $k_2 = 0.2$, $k_3 = 0.4$, $k_4 = 0.1$ and $k_h = 0.4$. The minimal steering angle rate k_3 has the highest priority in this experiment. The obtained result shows that the obtained waypoints are localized very close to the border of the road (cf. Fig. 4.30(b)). Fig. 4.30(a) and 4.30(b) show the actual vehicle's trajectories from the simulation and experiment. It can be observed that they are very close (maximal error between them is less than $0.15\ m$). We can conclude thus that the proposed optimal multi-criteria waypoint selection based on Expanding Tree (OMWS-ET, performed off-line (cf. Section 4.3.3.2)) permits to cope accurately with actual environment and experiments.

Fig. 4.29(c) and 4.30(c) show the comparison between the set of waypoints and the real trajectories of both experiments. The velocities and steering angle of the vehicle while

[1] http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/OMWS.mp4

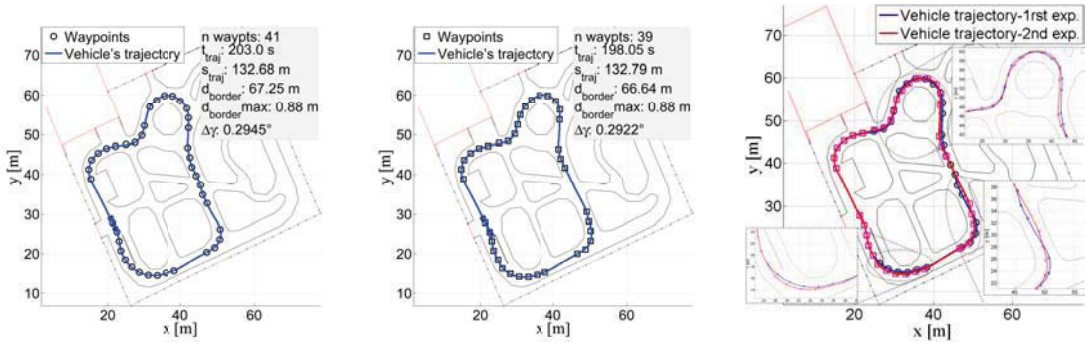


(a) First experiment: Safe planning.

(b) Second experiment: Steering angle minimization.

(c) Comparison between experiments.

Figure 4.29: Different set of obtained waypoints using OMWS-ET.



(a) First experiment.

(b) Second experiment.

(c) Comparison between actual experiments.

Figure 4.30: Actual vehicle's trajectories for different obtained set of waypoints.

reaching each waypoint are shown in Fig. 4.31.

Table 4.3 shows different performance criteria to compare the set of waypoints where: n_w is the number of waypoints, T is the navigation time, l_{UGV} is the traveled distance, d_{border} is the sum of minimum distance at each UGV's position to the road boundaries along the whole UGV's trajectory and $\Delta\gamma$ is the root mean square (rms) of the steering angle rate. It is noted that the first experiment has n_w greater than the second experiment. It is due to the fact that the first experiment has the safety as main priority. The proposed Algorithm 3 selects thus more waypoints to allows the vehicle to navigate as farther as

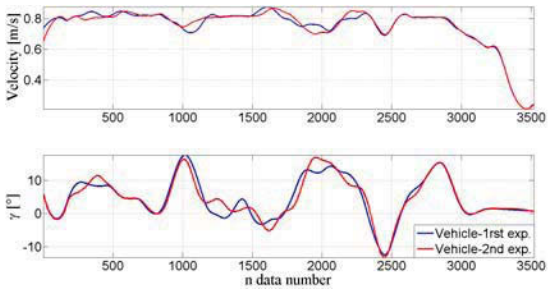


Figure 4.31: Vehicle's velocities and steering angles progress for each set of waypoints.

		n_w	$T[s]$	l_{UGV}[m]	d_{border}[m]	$\Delta\gamma[^\circ]$
1rst exp.	Sim.	41	200	132.81	67.35	0.3123
	Real	41	203	132.68	67.25	0.2945
2nd exp.	Sim.	39	199	133.00	66.54	0.3089
	Real	39	198	132.79	66.64	0.2922

Table 4.3: Comparison among the set of waypoints

possible from the road borders. It can be noticed by d_{border} where its value is bigger in the first experiment than the second. Furthermore, the values of $\Delta\gamma$ is less in the second experiment because the highest priority was for the steering angle rate. Therefore, the vehicle can navigate with higher velocity along the trajectory and the navigation time is smaller than the first experiment.

4.4.2/ SAFE AND FLEXIBLE NAVIGATION THROUGH SUCCESSIVE WAYPOINTS

This subsection focuses on the performance of the control architecture with the proposed target assignment strategy (cf. Section 4.2) using waypoint selection (OMWS) based on Algorithm 3 in the well-know environment (PAVIN) (cf. Fig. 4.32). The scenario was built to exhibit the flexibility of the proposed control architecture for obstacle avoidance situation (unexpected obstacle in the environment) (cf. Fig. 4.32 (c) and (f)). Two obstacle are placed between the waypoints (cf. Fig. 4.32 (b) and (e)). As mentioned in Section 4.2, the proposed control architecture integrates the obstacle avoidance behavior (cf. Fig. 4.4). Therefore, the vehicle can perform different maneuvers between waypoints, in this case the obstacle avoidance without the use of any trajectory replanning method. The elliptical limit-cycle trajectories are applied for reactive obstacle avoidance using only position information and uncertain range data (cf. Section 3.2). Each hinder obstacle is surrounded by an ellipse which parameters are computed online using range data (cf. Section 3.4). The UGV has to reach successively each static waypoint while avoiding hinder obstacles. This experiment can be found online[2].

[2]http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/NavigationWaypts.mp4

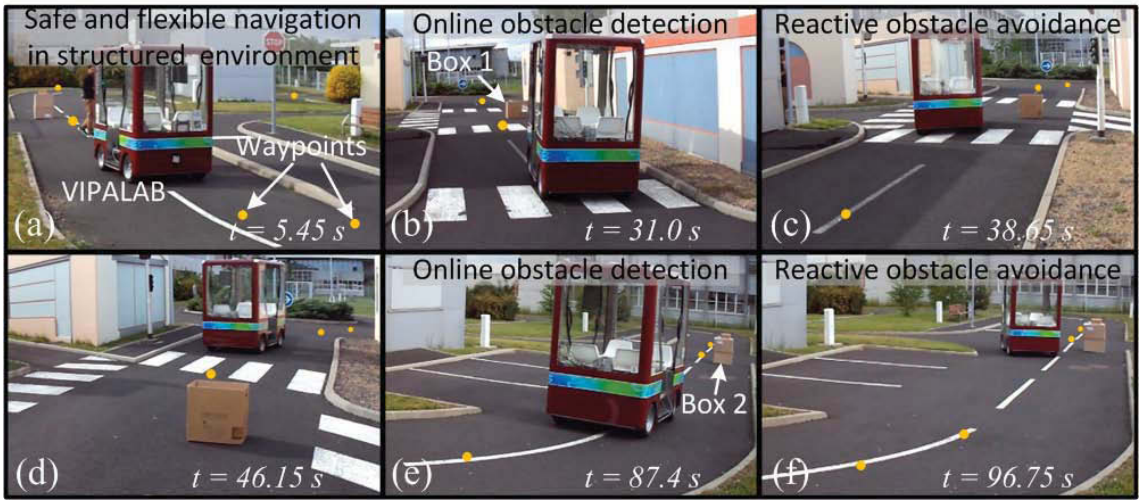
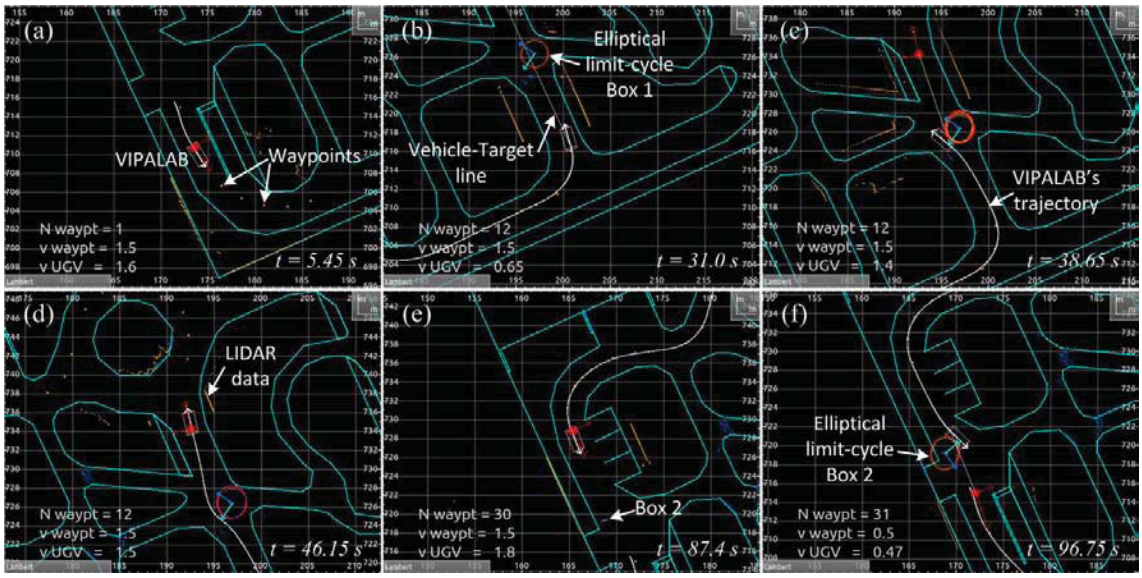
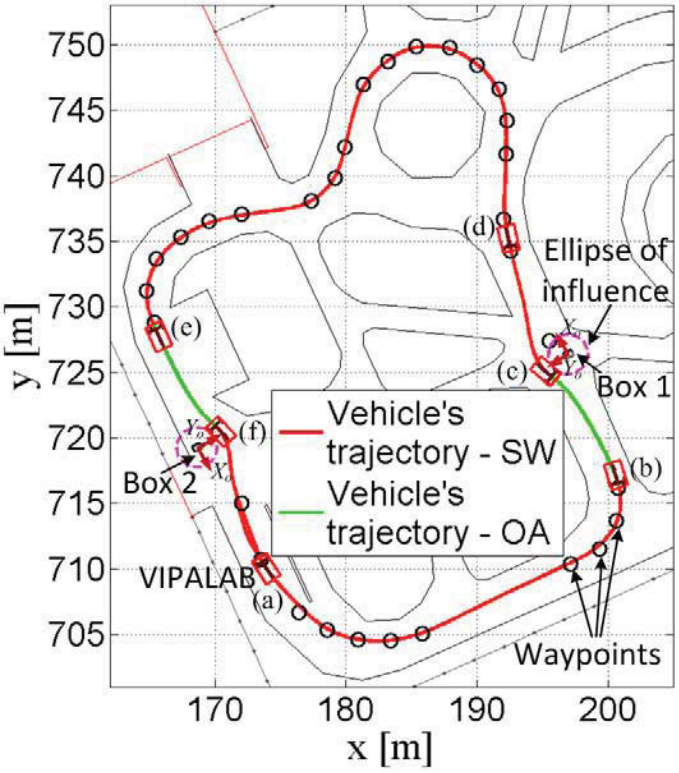


Figure 4.32: Safe and flexible navigation through successive waypoints .

Some screenshots of the developed Graphical Data Interface for VIPALAB (GDI-VIPA) are shown in Fig. 4.33(a). The white line represent UGV's trajectory, the orange points



(a) GDI-VIPA during the vehicle's navigation ("v UGV" and "v waypoints" are respectively the current velocities of the UGV and the waypoint ("N waypoint")).



(b) Vehicle's trajectory using the proposed control architecture and a set of waypoints positioned in the environment using Algorithm 3 (SW: Successive Waypoints and OA: Obstacle Avoidance).

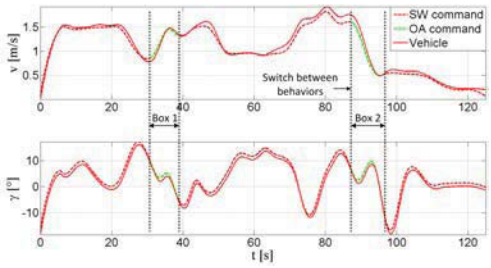
Figure 4.33: Validation of the navigation through successive waypoints.

are the set of waypoints and the big red point is the current assigned waypoint (target). It can be noted the online detection of the boxes using the LIDAR sensor (cf. Section 3.4) and the reactive obstacle avoidance (based on elliptical limit-cycle (cf. Section 3.2)) performed by the UGV (cf. Fig. 4.33(a) (b), (c) and (f)). Therefore, the vehicle can perform different maneuvers between waypoints, in this case the obstacle avoidance without the use of any trajectory replanning method.

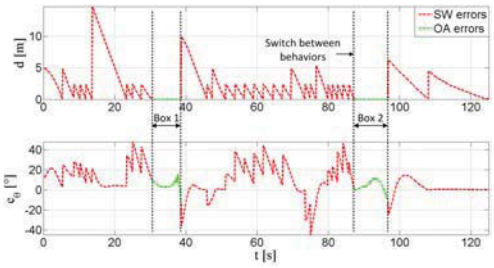
Fig. 4.33(b) shows that the UGV reaches accurately the successive static waypoints. Moreover, this figure shows the vehicle's trajectories when the obstacle avoidance is activated (green line). The UGV detects the hinder obstacle between the waypoints and it applies the reactive limit-cycle method [2, 3]. The UGV does never collide with any obstacle.

Fig. 4.34(a) shows the velocity and steering angle of the vehicle. It can be observed that the UGV's velocity changes according to waypoint's velocity and decreases in the obstacle avoidance phase (since the velocity set-point in this phase takes into account the distance to the obstacle (cf. Section 3.2)) which allows to obtain a smooth and safe navigation. Fig. 4.34(b) shows the controller parameters (position and orientation errors). It is noted that the absolute value of errors converges towards zero for each waypoints. Therefore, the control law given by (3.23 on page 62) and (3.24 on page 62) drives with reliable and stable way the vehicle towards the assigned set of waypoints. Fig. 4.34(c) show the Lyapunov function values which highlight that the vehicle converges to each static waypoint.

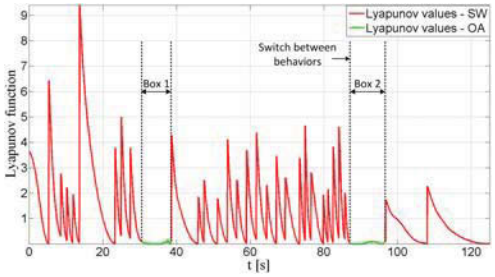
Therefore, the proposed navigation strategy allows safe, flexible and smooth trajectories between the waypoints and also to perform different behaviors, such as obstacle avoidance.



(a) Commands given by the control law.



(b) Position and orientation errors of the vehicle.



(c) Lyapunov function of the vehicle.

Figure 4.34: Experimental results of the safe and flexible navigation through successive waypoints (SW: Sequential Waypoint and OA: Obstacle Avoidance).

4.5/ CONCLUSION

This chapter presented an overall control architecture to cope with the problem of navigation in a structured environment. A novel safe and flexible navigation strategy based on successive target reaching for a UGV in structured environment was presented. It is based on target switching using appropriate reference frames, linked to the current selected waypoint and to the next one in the list. This target assignment strategy enables a smooth and flexible vehicle trajectory while satisfying an upper bound on distance and orientation errors (to not collide with the boundary of the environment). Furthermore, different target-reaching methods from the literature were presented and their performances were compared with the proposed navigation strategy. These comparisons showed the interesting features of the proposed control law in term of stability and flexibility for different tasks. Simulations shows the efficiency and the flexibility of the proposed navigation strategy for an urban vehicle in structured environment. Furthermore, this chapter has presented three planning methods to obtain the optimal waypoints configuration (Primary waypoint selection, Optimal Multi-criteria Waypoints Selection based on Expanding Tree (OMWS-ET) and Grid Map (OMWS-GM)) which guarantees safe, smooth and feasible vehicle navigation in a structured environment. The proposed OMWS-GM is based on the A^* algorithm with an additional term to consider the orientation change between successive cells. OMWS-ET uses a multi-criteria function which takes into account the vehicle model and localization uncertainties to obtain the optimal set of waypoint configurations (position, orientation and velocity). Moreover, it has been shown that the proposed OMWS-ET is more accurate and flexible than OMWS-GM. Several simulations demonstrate the efficiency and reliability of the proposed OMWS-ET in different cases (trajectory specification, deterministic versus probabilistic, comparison with RRT*, multi-robot formation, local replanning according to the multi-criteria optimization).

5

MULTI-VEHICLE NAVIGATION IN FORMATION

Control and coordination of multiple robots is a challenging domain which has increased significantly over the last decades. Different tasks that may be performed by a single complex robot can be performed with more flexibility and efficiency by a group of elementary cooperative robots. Some examples of multi-robot tasks can be found in Section 2.1 such as: exploration [329], mapping of unknown locations [237], coverage of unknown area [296], transport of heavy objects [382] and platooning for public transportation [283]. This chapter addresses the navigation of a group of vehicles in formation with reconfiguration (cf. Section 2.3), i.e., when a group of mobile robots has to navigate while keeping (or changing) the desired relative positions to each other according to the context of the environment.

Different reconfiguration methods from the literature are presented in the next section. Section 5.2 presents the proposed navigation in formation based on *Leader-follower* and *Behavior-based* approaches according to the formation frame. An analysis of the formation constraints (physics, geometric and kinematics) is detailed in Section 5.3. Section 5.4 presents the proposed reconfiguration methods for the multi-robot dynamic formation. Several simulations and experiments are performed to demonstrate the stability, reliability, flexibility and advantages of the navigation in formation of a group of UGVs for each section. Finally, Section 5.5 provides a conclusion.

Figure 5.1: Autonomous navigation in formation of a group of UGVs in an urban environment (Clermont-Ferrand, France).

5.1/ RELATED WORKS

In Section 2.3, several approaches have been described to deal with formation control problem such as: *behavior-based* [224], *virtual structure* [127] and *Leader-follower* [281], etc. This chapter proposes a new control architecture based on *Leader-follower* and *behavior-based* approaches for the formation control problem.

The case of dynamic formation or reconfiguration, i.e., the formation shape changes to another (e.g. from square to triangle) was dealt in [142, 210, 311]. In [210], the leader generates a free-collision trajectory in a dynamic environment which is tracked using a formation control law based on neural network (NN), Lyapunov function and dynamic model of the robot. The stability of the dynamic formation and dynamic topology (adjacency matrix) are also demonstrated. In [142], switches between different formation shapes are exploited (from triangle to line) to avoid encountered obstacles in the environment. The formation control law is based on input-output feedback linearization and vision sensors (omnidirectional camera) are embedded in each robot for localization and navigation purpose. A strategy to modify the formation configuration by scaling the distance between the vehicles is proposed in [311]. Obstacle avoidance for the formation is dealt while using potential fields.

In [164], the authors added complex maneuvers that are decomposed into a sequence of elementary maneuvers between formation patterns (desired position of the robots). The behavior-based formation control relies on coupled dynamics (relative position and velocity) between neighbor robots. A non-rigid formation control that determines the most appropriate robot positions (formation shape) according to the directional visual perception was proposed in [177]. The inter-robot communication allows to choose the leader of the group. The formation control with collision avoidance based on behavioral control was proposed in [208]. The authors propose a Null-spaced-based behavioral control to coordinate the formation while performing different tasks. These approaches are very flexible and extensible to more complex tasks, nevertheless, it is difficult to guaranty the stability of the overall system. A static and dynamic virtual structure based on Lyapunov approach was proposed in [248]. A virtual structure with obstacle avoidance for a group of UAV was proposed in [321]. The approach uses a penalty function (distance to the virtual target, obstacle) and a priority strategy (according to the distance to an object (other UAV or an obstacle). The control law use the Model Predictive Control (MPC) based on time horizon and optimization of a cost function. Nevertheless, this method is generally time consuming due to predictive computation w.r.t. a time horizon.

In the already described works, interesting solutions for formation control problem are proposed. Nevertheless, they are based generally on predefined trajectories and they do not address issues related to constraints of the formation shape and to the UGV's kinematics (large UGVs require large space for navigation and obstacle avoidance). This chapter describes the navigation of a group of UGVs while keeping a specific formation and reconfiguration according to the environmental context. The dynamics of the followers' set-points are given by the specific dynamic of the leader (*Leader-follower* approach). This strategy allows a good flexibility properties for the formation shape [249] (cf. Section 5.2). The used *behavior-based* approach allows to use different elementary controllers to perform sequentially several sub-tasks (cf. Fig. 5.2). The dynamic of the formation according to vehicles' constraints is analyzed. A method to adapt the formation constraints, velocities and steering angle, according to the dynamic of the leader is proposed. The

convergence of the formation is guaranteed when the adaptive constraints are satisfied (cf. Section 5.3). A new Strategy for Formation Reconfiguration (SFR) of the group of UGVs based on suitable smooth adaptation of the set-points (according, for instance, to the encountered obstacles or the new task to achieve) is also proposed in Section 5.4.

5.2/ PROPOSED STRATEGY FOR NAVIGATION IN FORMATION

The objective of the group of N UGVs is to reach and to keep their assigned configuration according to the desired formation shape and leader's configuration [142, 345]. The proposed strategy consists in controlling each UGV (follower) to track its assigned virtual dynamic target [5] (cf. Subsection 5.2.1 and 5.2.2). This strategy, based on *Leader-follower* and *Behavior-based* approaches, is defined by:

- A leader (UGV$_L$ in Fig. 5.3 and 5.4); its pose (x_L, y_L, θ_L) and its linear velocity v_L determine the dynamic of the formation (cf. Fig. 5.3).

- The formation structure is defined with as much nodes as necessary to obtain the desired formation shape. Each node i is a virtual dynamic target (T_{d_i}). The formation is defined as $\mathbf{F} = \{\mathbf{f}_i, i = 1 \cdots N\}$, where $\mathbf{f}_i$ are the coordinates $(h_i, l_i)^T$ of the dynamic target T_{d_i} w.r.t. the leader's local reference frame (cf. Fig. 5.3 and 5.4).

The proposed control architecture for the navigation in formation (cf. Fig. 5.2 where $N = 2$) is based on the one proposed in Section 4.3 for the navigation through successive waypoints in structured environment. At this aim, two new blocks, *Communications* and *Formation parameters* have been added. The *Communications* block is related with the UGV's capability to send and to receive information from other UGVs. The *Formation parameters* block is designed to generate the virtual dynamic target T_{d_i} that each follower has to track using the proposed target-reaching control [4, 8] (cf. Section 3.5).

An important advantage of the proposed strategy for navigation in formation is that it takes, in addition to the target's positions (x_T, y_T), the heading θ_T of the virtual targets, which allows to have accurate formation navigation (cf. Section 5.2.3). Furthermore, the proposed approach is flexible in the sense that it does not generally depend on any pre-defined reference trajectory [210, 311]. The formation is fully defined by the instantaneous dynamic of the leader.

One important consideration to achieve the proposed Multi-Robot Formation (MRF), is that the followers have to know, as accurately as possible, the leader state (pose and velocity). The leader sends its state by stable Wi-Fi communication (cf. Fig. 5.2). However, cameras and/or LIDAR sensors embedded in each follower, can be used to estimate the leader's state [142, 323].

A detailed survey of two suitable frames for formation's definition, Cartesian and Frenet frames, are presented in next subsections. In the sequel, $\mathbf{f}_i$ is given in Global Cartesian frame to homogenize the notation of the presented approaches (cf. Subsections 5.2.1 and 5.2.2).

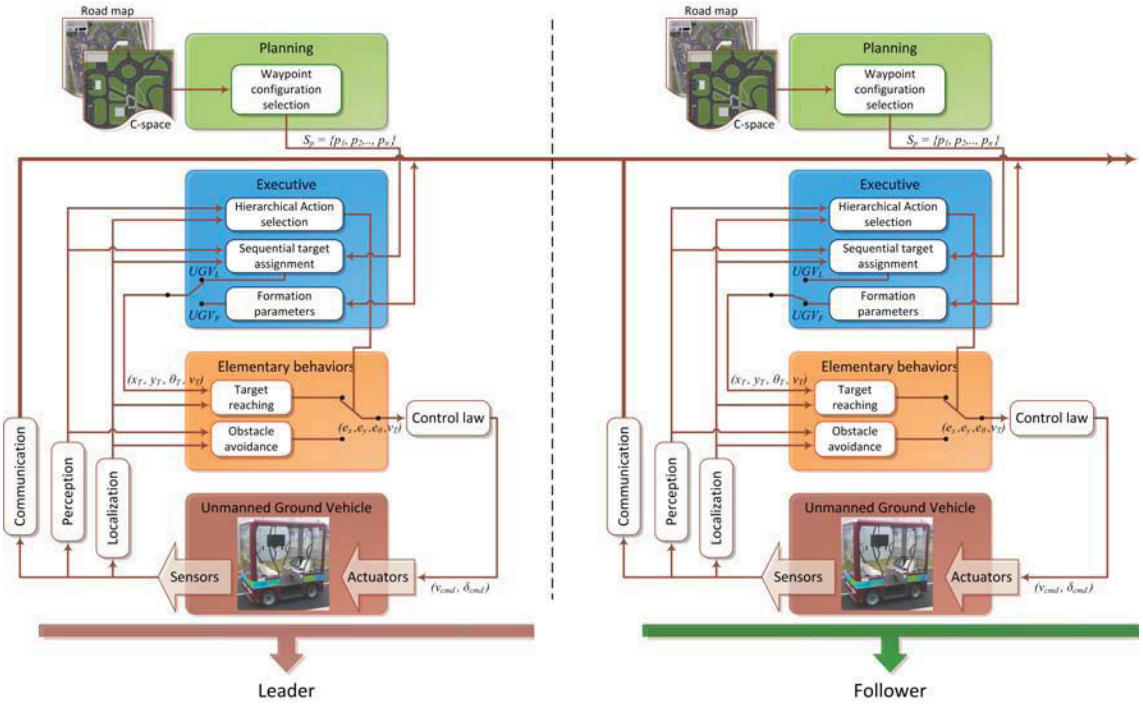


Figure 5.2: Proposed control architecture for navigation in formation of a group of UGVs.

5.2.1/ CARTESIAN FORMATION

In this subsection, a rigid definition of the formation is applied to maintain the formation shape (e.g. a triangle in Fig. 5.3). The position and orientation of each node (virtual target) are computed from the leader's configuration w.r.t. the Cartesian frame (local frame of the leader $X_m Y_m$) [236, 278, 311]. The leader's position determines the nodes' positions according to the formation shape. The instantaneous center of curvature Icc_L of the formation is determined by the leader according to its movements (cf. Fig. 5.3). Icc_L allows to compute the desired orientation of the nodes according to the formation shape (cf. Fig. 5.3). The leader turns around Icc_L (positioned perpendicularly to its rear wheel), then the other target set-points T_{d_i} must also turn around Icc_L to maintain the rigid formation. Thus, the target velocity v_{T_i} must be tangent to the circle which has Icc_L as center and the distance between T_{d_i} and Icc_L as radius r_{cT_i}.

The idea behind this strategy is to eliminate the dependency of each UGV to a global reference frame. A straightforward transformation can be applied to obtain the set-point w.r.t. a local reference frame attached to the leader. The polar coordinates (l_i, ψ_i) can also be used by applying an elementary transformation [236, 278].

The pose of the virtual target T_{d_i} w.r.t the leader pose in the Global reference frame can be written as (cf. Fig. 5.3):

$$\begin{bmatrix} x^c_{T_i} \\ y^c_{T_i} \\ \theta^c_{T_i} \end{bmatrix} = \begin{bmatrix} x_L \\ y_L \\ \theta_L \end{bmatrix} + \begin{bmatrix} \cos(\theta_L) & -\sin(\theta_L) & 0 \\ \sin(\theta_L) & \cos(\theta_L) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h^c_i \\ l^c_i \\ \beta_i \end{bmatrix} \tag{5.1}$$

where (x_L, y_L, θ_L) is the current pose of the leader and β_i is the T_{d_i} orientation w.r.t. the leader's pose. (h^c_i, l^c_i) are the coordinates of T_{d_i} w.r.t. mobile Cartesian frame $X_m Y_m$. It is

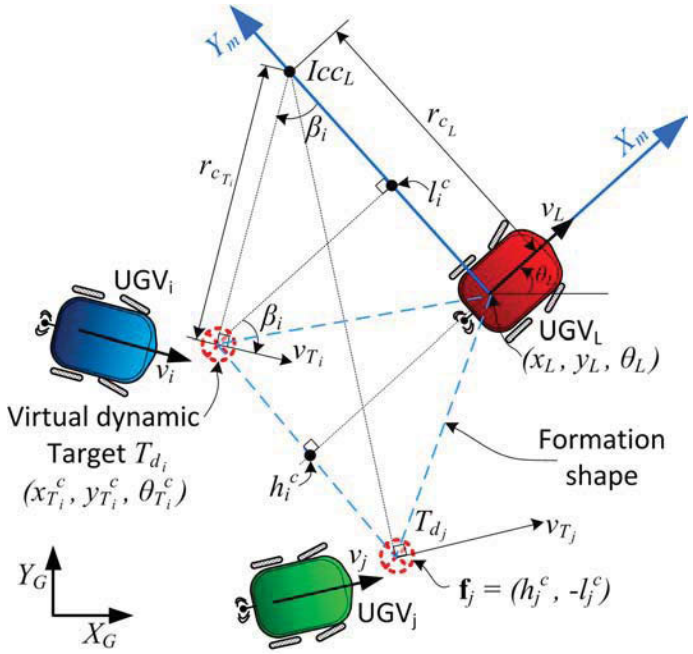


Figure 5.3: Formation definition in mobile Cartesian frame linked to the Leader.

given by:

$$\beta_i = \arctan\left(\frac{h_i^c}{r_{c_L} - l_i^c}\right) \tag{5.2}$$

where r_{c_L} is the radius of curvature of the leader. Differentiating eq. (5.1), the velocities of each T_{d_i} are thus given by:

$$v_{T_i}^c = \sqrt{(v_L - l_i^c \omega_L)^2 + (h_i^c \omega_L)^2} \tag{5.3}$$

$$\omega_{T_i}^c = \omega_L + \dot{\beta}_i \tag{5.4}$$

where v_L and ω_L are respectively the linear and angular velocities of the leader, $\dot{\beta}_i$ is computed as:

$$\dot{\beta}_i = \frac{-h_i^c \dot{r}_{c_L}}{(r_{c_L} - l_i^c)^2 + (h_i^c)^2} \tag{5.5}$$

One can note from eq. (5.5) that when $\dot{\beta}_i$ is equal to zero, the formation has a constant radius of curvature r_{c_L} and the angular velocities of the virtual targets are equal to the angular velocity of the leader ($\omega_{T_i}^c = \omega_L$) eq. (5.4).

5.2.2/ FRENET FORMATION

This approach is applied when tracking the leader's movements is more important than keeping a rigid shape during the navigation in very structured environments (limited roads). The objective of this Frenet frame is to adapt the formation to the leader's trajectory. This trajectory is used to define the formation in longitudinal (curvilinear) and lateral (perpendicular to the trajectory) coordinates [277, 302] (cf. Fig. 5.4). In this section, $\mathbf{f}_i^f = (h_i^f, l_i^f)^T$ is given in Frenet coordinates where (h_i^f, l_i^f) are respectively the curvilinear

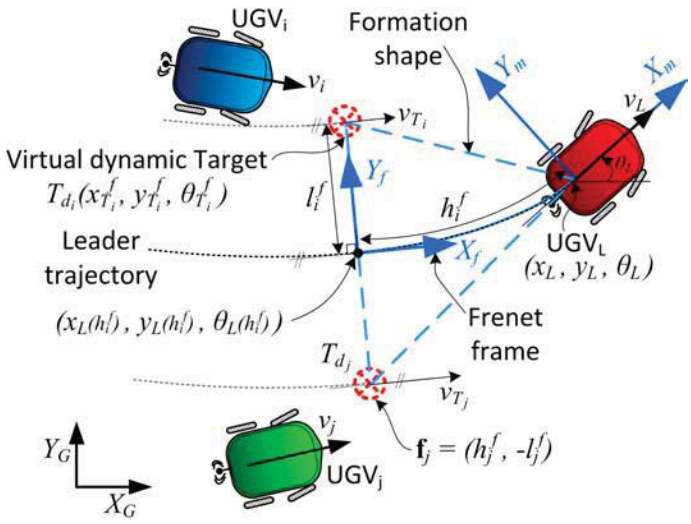


Figure 5.4: Formation definition in Frenet frame linked to the Leader's trajectory.

and lateral distance w.r.t. the leader trajectory. The posture of the virtual target T_{d_i} w.r.t. the leader trajectory given in the Global reference frame can be written as:

$$\begin{bmatrix} x_{T_i}^f \\ y_{T_i}^f \\ \theta_{T_i}^f \end{bmatrix} = \begin{bmatrix} x_L(h_i^f) \\ y_L(h_i^f) \\ \theta_L(h_i^f) \end{bmatrix} + \begin{bmatrix} \cos(\theta_L(h_i^f)) & -\sin(\theta_L(h_i^f)) & 0 \\ \sin(\theta_L(h_i^f)) & \cos(\theta_L(h_i^f)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ l_i^f \\ 0 \end{bmatrix} \tag{5.6}$$

where $(x_L(h_i^f), y_L(h_i^f), \theta_L(h_i^f))$ is the pose of the leader at h_i longitudinal distance from the current leader's pose along its trajectory (cf. Fig. 5.4). If $h_i^f < 0$ then T_{d_i} is back to the current leader's pose.

The coordinates h_i^f has to be constant to keep the desired curvilinear distance then, the linear velocity at h_i^f must be equal to current linear velocity of the leader ($v_L(h_i^f) = v_L$). The velocities of each T_{d_i} are given by:

$$v_{T_i}^f = v_L - l_i^f \omega_L(h_i^f) \tag{5.7}$$

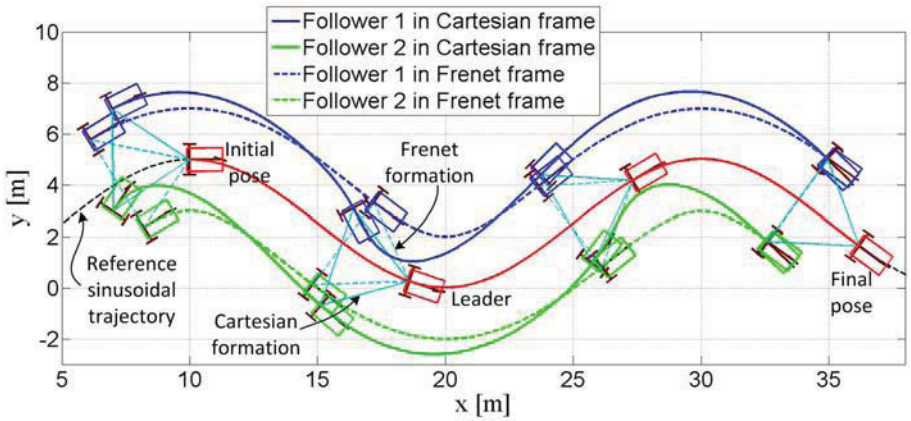
$$\omega_{T_i}^f = \omega_L(h_i^f) = \frac{v_L}{r_{c_L}(h_i^f)} \tag{5.8}$$

where $r_{c_L}(h_i^f)$ is the radius of curvature at h_i^f (longitudinal distance from the current pose of the leader along its trajectory).

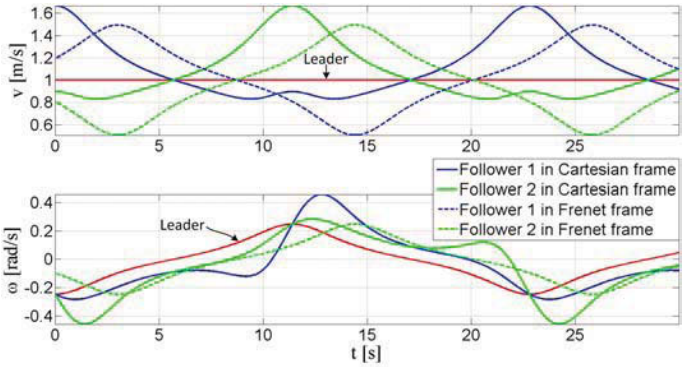
5.2.3/ Simulation analysis of the navigation in formation

This section shows the evolution of a group of $N = 3$ UGVs that navigates while keeping the desired formation. Simulations using tricycle vehicles shows the efficiency and the flexibility of the proposed navigation in formation using only the leader's dynamic.

In these simulation, the used UGV's parameters, constraints, sensors and controller parameters were described in Section 3.6. In addition for multi-vehicle navigation, each UGV has a stable communication network between UGVs without latency.



(a) Vehicles' trajectories.



(b) Vehicles' velocities.

Figure 5.5: Example of navigation in formation using the Cartesian and Frenet frames.

Firstly, to illustrate the application of each formation frame, a navigation for two presented formation definition with a constant linear velocity of the leader ($v_L = 1\ m/s$) was simulated. Fig. 5.5(a) shows the navigation for the two formation frames. It can be observed that the Frenet formation has parallel trajectories w.r.t. the leader's trajectory, however the geometric shape changes along the path (dashed lines). The Cartesian formation keeps the geometric shape along the path, but its trajectory has a variable deviation from the leader's trajectory. The velocities of the UGVs are shown in Fig. 5.5(b). It can be noted that the linear and angular velocities of the Cartesian Formation (continuous lines) are greater than the Frenet formation (dashed lines), mainly during curves, because the velocities of the virtual targets of the Cartesian formation are increased to keep the geometric shape along the leader's trajectory (5.3). Moreover, the angular velocities of the Frenet formation are equal to the leader with an offset. It happens when the leader velocity is constant, thus, the angular velocities of the virtual targets of the Frenet formation depends only on radius of curvature of the leader at h_i^f along the leader trajectory (5.8).

5.2.3.1/ CARTESIAN FORMATION ANALYSIS

To analyze the dynamic of the Cartesian formation in function of the leader dynamic, three different formation shapes $\mathbf{F} = \mathbf{F}_1,\ \mathbf{F}_2,\ \mathbf{F}_3$ are used for three different leader velocities $v = 1.0,\ 1.5,\ 2.0\ m/s$, where $\mathbf{F}_1 = \{\mathbf{f}_1(-2,-1.5),\ \mathbf{f}_2(-2,1.5)\}\ m$, $\mathbf{F}_2 = \{\mathbf{f}_1(-4,-3),\ \mathbf{f}_2(-4,3)\}\ m$

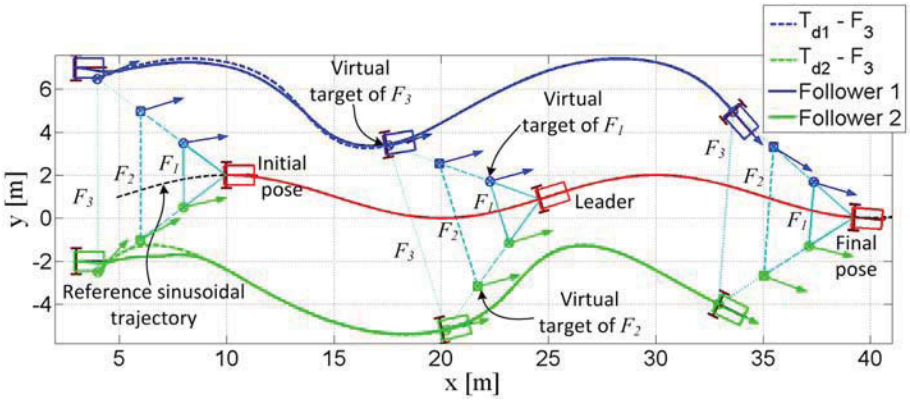
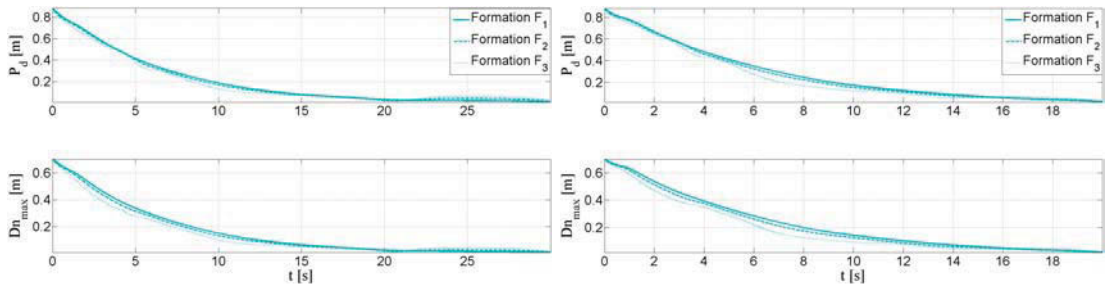


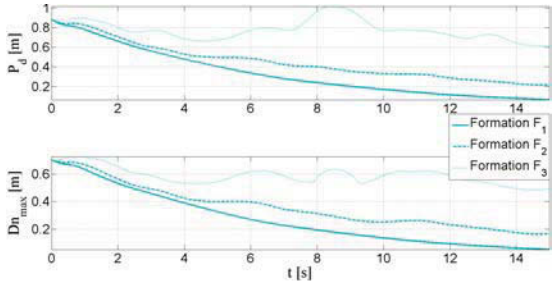
Figure 5.6: Schema of comparison of between Cartesian formations $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ ($v_L = 1.5\ m/s$) ($T_{d_i} - F3$ is the i dynamic virtual target w.r.t. $\mathbf{F}_3$ formation).

and $\mathbf{F}_3 = \{\mathbf{f}_1(-6, -4.5),\ \mathbf{f}_2(-6, 4.5)\}\ m$. The leader makes a sinusoidal trajectory with an amplitude of $1\ m$ and a frequency of $20\ m$. Each follower tracks its virtual target to keep the desired formation $\mathbf{F}$ (cf. Fig. 5.6). The Procrustes shape distance is used as criteria to quantify the distortion between the desired formation $\mathbf{F}$ and the real formation shape (cf. Appendix F)

Fig. 5.7(a), 5.7(b) and 5.7(c) show the evolution of the Procrustes distance P_d and the maximum distance Dn_{max} between the positions of the vertex for different leader velocities $v_L = 1.0,\ 1.5$ and $2.0\ m/s$. The initial position of the vehicles has an offset $(1,\ 0.5)\ m$ from the initial position of its virtual targets. Fig. 5.6 shows the followers' trajectories to reach the formation $\mathbf{F}_3$. At the beginning of Fig. 5.7(a) ($v_L = 1.0\ m/s$) and 5.7(b) ($v_L = 1.5\ m/s$), the values of P_d and Dn_{max} of $\mathbf{F}_2$ and $\mathbf{F}_3$ are less than for $\mathbf{F}_1$. It occurs because the



(a) $v_L = 1.0\ m/s$ (b) $v_L = 1.5\ m/s$



(c) $v_L = 2.0\ m/s$

Figure 5.7: Cartesian formations $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ with different leader's velocities.

virtual target velocities depends on the formation dimension (5.3), thus, the virtual target velocities of $\mathbf{F}_2$ and $\mathbf{F}_3$ are greater than $\mathbf{F}_1$. The followers of $\mathbf{F}_2$ and $\mathbf{F}_3$ apply greater velocities commands (less than v_{max}) to keep the desired formation. All followers of the formation converges to $\mathbf{F}$ for low velocities. Nevertheless, Fig. 5.7(c) ($v_L = 2.0\ m/s$) shows the case where followers' velocity of $\mathbf{F}_2$ and $\mathbf{F}_3$ are saturated (virtual target velocity greater than v_{max}), thus, followers can not reach the desired formation $\mathbf{F}_2$ and $\mathbf{F}_3$. The values of P_d and Dn_{max} of $\mathbf{F}_2$ and $\mathbf{F}_3$ are greater than $\mathbf{F}_1$.

Table 5.1 shows the values of the $\mathcal{L}_2$ (cf. Appendix F) norm applied to the evolution of P_d and Dn_{max}. This table quantifies the analysis described above of Fig. 5.7(a), 5.7(b) and 5.7(c). The values of $\mathcal{L}_2(P_d)$ and $\mathcal{L}_2(Dn_{max})$ for different shapes are closer for low velocities $v_L = 1.0$ and $1.5\ m/s$. Nevertheless, the saturation occurs when $v_L = 2.0\ m/s$ and the values of $\mathcal{L}_2(P_d)$ and $\mathcal{L}_2(Dn_{max})$ of $\mathbf{F}_2$ and $\mathbf{F}_3$ are greater than $\mathbf{F}_1$. This table confirm that a steady formation shape depends on the formation dimension and the leader's dynamic.

	$\mathcal{L}_2(P_d)\ [m]$			$\mathcal{L}_2(Dn_{max})\ [m]$		
$v[m/s]$	1.0	1.5	2.0	1.0	1.5	2.0
$\mathbf{F}_1$	0.1899	0.2643	0.3201	0.1525	0.2133	0.2589
$\mathbf{F}_2$	0.1857	0.2591	0.3956	0.1458	0.2043	0.3174
$\mathbf{F}_3$	0.1811	0.2520	0.6475	0.1401	0.1917	0.4782

Table 5.1: Procrustes distances and maximum distance for different velocities and dimensions.

5.2.3.2/ FRENET FORMATION ANALYSIS

This subsection analyzes the dynamic of the Frenet formation in function of the leader trajectory. At this aim, three different formation shapes $\mathbf{F} = \mathbf{F}_1,\ \mathbf{F}_2,\ \mathbf{F}_3$ are used for three different leader velocities $v = 1.0,\ 1.5,\ 2.0\ m/s$, where $\mathbf{F}_1 = \{\mathbf{f}_1(-2,-1.5),\ \mathbf{f}_2(-2,1.5)\}\ m$, $\mathbf{F}_2 = \{\mathbf{f}_1(-2,-3),\ \mathbf{f}_2(-2,3)\}\ m$ and $\mathbf{F}_3 = \{\mathbf{f}_1(-2,-4.5),\ \mathbf{f}_2(-2,4.5)\}\ m$.

The group of UGVs (leader and followers) performs the same scenario as in Subsection 5.2.3.1 (cf. Fig. 5.8). The distance and orientation errors of each follower w.r.t. its target are used as criteria to quantify the performance of the formation $\mathbf{F}$ such as:

$$d_{rms} = \frac{1}{N}\sqrt{\sum_{i=1}^{N} d_i^2} \tag{5.9}$$

$$e_{\theta_{rms}} = \frac{1}{N}\sqrt{\sum_{i=1}^{N} e_{\theta_i}^2} \tag{5.10}$$

where N is the number of UGVs. Fig. 5.8 shows the followers' trajectories to reach the formation $\mathbf{F}_3$. The evolutions of the distance d_{rms} and orientation $e_{\theta_{rms}}$ errors for different leader velocities $v_L = 1.0,\ 1.5$ and $2.0\ m/s$ are shown in Fig. 5.9.

Fig. 5.9(a) ($v_L = 1.0\ m/s$) and 5.9(b) ($v_L = 1.5\ m/s$) shows that the UGVs reach their assigned target for each different formation shape. Nevertheless, Fig. 5.9(c) ($v_L = 2.0\ m/s$) shows that the UGV followers (as in the case of Cartesian formation for $\mathbf{F}_2$ and $\mathbf{F}_3$) have saturated velocities since the virtual target's velocity is greater than v_{max}. Then, the followers can not reach their assigned targets. The values of d_{rms} and $e_{\theta_{rms}}$ of $\mathbf{F}_2$ and $\mathbf{F}_3$ are greater than $\mathbf{F}_1$.

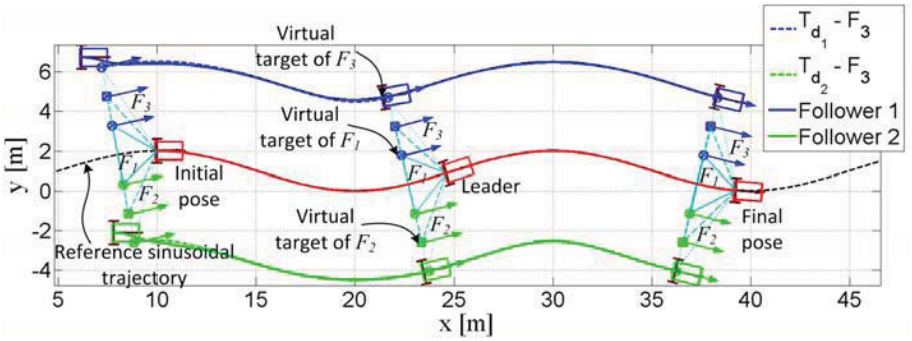
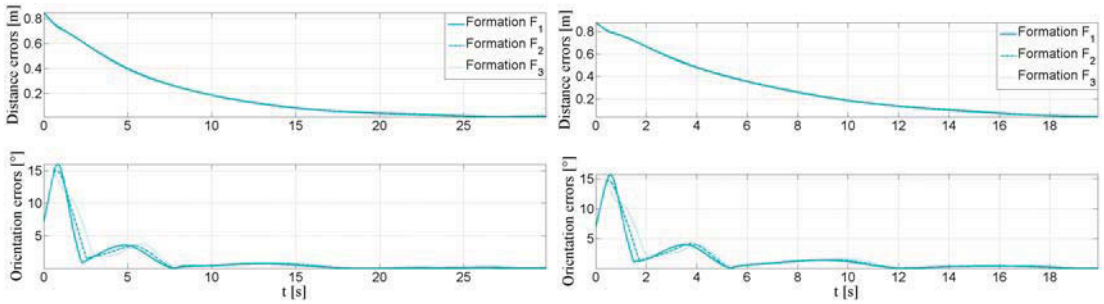


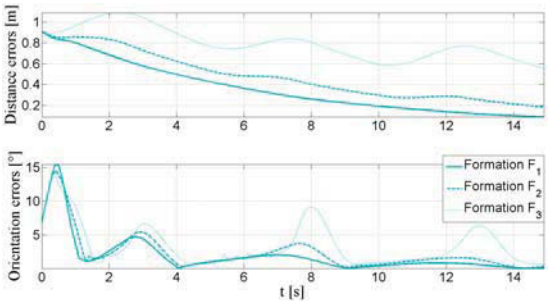
Figure 5.8: Schema of comparison of between Frenet formations $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ ($v_L = 1.5\ m/s$) ($T_{d_i} - F3$ is the i dynamic virtual target w.r.t. $\mathbf{F}_3$ formation).

Table 5.2 quantify the analysis of Fig. 5.9(a), 5.9(b) and 5.9(c) using the $\mathcal{L}_2$ (cf. Appendix F) norm applied to the evolution of d_{rms} and $e_{\theta_{rms}}$.

As expected from Cartesian formation analysis (cf. Subsection 5.2.3.1), the values of $\mathcal{L}_2(d_{rms})$ and $\mathcal{L}_2(e_{\theta_{rms}})$ for different shapes are closer for low velocities $v_L = 1.0$ and $1.5\ m/s$. Nevertheless, when the velocity is high ($v_L = 2.0\ m/s$), the values of $\mathcal{L}_2(d_{rms})$ and $\mathcal{L}_2(e_{\theta_{rms}})$ of $\mathbf{F}_2$ and $\mathbf{F}_3$ are greater than $\mathbf{F}_1$ since the saturation occurs for great formation shape. This tables confirm that a steady formation w.r.t. leader's trajectory depends on the formation dimension and the leader's dynamic.



(a) $v_L = 1.0\ m/s$ (b) $v_L = 1.5\ m/s$



(c) $v_L = 2.0\ m/s$

Figure 5.9: Frenet formations $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ with different leader's velocities.

	$\mathcal{L}_2(d_{rms})$ $[m]$			$\mathcal{L}_2(e_{\theta_{rms}})$ $[^\circ]$		
$v[m/s]$	1.0	1.5	2.0	1.0	1.5	2.0
$\mathbf{F}_1$	0.2835	3.2239	0.3591	3.2879	0.4272	3.3738
$\mathbf{F}_2$	0.2845	3.2804	0.3604	3.3487	0.5259	3.6247
$\mathbf{F}_3$	0.2863	3.3916	0.3624	3.4251	0.8082	4.5671

Table 5.2: Distances and orientation errors for different velocities and dimensions.

5.3/ FORMATION CONSTRAINTS

Different applications require that the group of UGVs have to accurately keep the formation shape during the navigation (e.g. transportation and reposition of heavy objects [175], exploration [329], platooning of military vehicle [238], etc.). At this aim, the dynamic of leader must be constrained in order to allow each UGV to always converges to its assigned target. This section analyzes the dynamic of the formation according to UGVs' constraints.

In Section 2.3, several works presented interesting solutions for formation control problem. Nonetheless, they have not analyzed the constraints of the formation shape and the capabilities to be scaled to large vehicles (which need large space for navigation and obstacle avoidance). In [92], the authors present an algorithm to minimize the deformation of the rigid structure for rotational and translational displacement of the rigid structure outside its reachable area. In [232], optimal trajectories were used to guide the followers according to the leader's trajectory. However, the proposed method requires an appropriate optimal trajectory planning which needs high computation for environments with large number of obstacles. In [249], the authors analyze the geometry of the formation to impose a bound on the curvature of the leader's trajectory. The formation control using input constraints is designed to adapt the formation geometry to the follower dynamic (the follower positions vary in suitable cones) to obtain lower control effort w.r.t. approaches based on rigid formation. However, in this work, the distortion of formation shape is not quantified. A criteria to evaluate the formation shape based on Procrustes distance was proposed in [345]. The authors use a virtual *Leader-follower* approach with artificial potential fields and the dynamic model of the robot. In [350], attainable set-points for obstacle avoidance behavior are generated according to the kinematic constraints of a unicycle robot and the radius of the limit-cycle (cf. Section 3.2). Nevertheless, the authors did not consider neither the dimension of the formation nor the steady convergence toward the desired formation geometric shape.

In this section is proposed an adaptive computation of the leader's constraints (its maximum linear velocity and steering angle (or radius of curvature)) to obtain the dynamic of each virtual target (thus the set-points for the followers) which satisfy the vehicles' constraints of the overall formation. The adaptive leader's constraints are related to its dynamic and the geometrical shape of the formation. These adaptive constraints allow to improve the convergence of the formation shape and to keep it during the multi-robot navigation (cf. Section 5.2.3).

5.3.1/ CONSTRAINTS IN CARTESIAN FORMATION

The local frame of the leader allows to keep a constant geometric structure during the navigation of the group of UGVs (cf. Section 5.2.3). The dynamic of this geometric structure is subordinated to the dynamic of the leader (cf. eq. (5.1), (5.3) and (5.4)). It is advantageous because the followers only require leader's configuration to determine its desired position in the formation [210]. Nevertheless, the dynamic of the leader has to be more constrained than the followers according to the formation shape to satisfy the followers' dynamic constraints [211, 249], i.e., the dynamic of the virtual target can be greater than its maximum values (5.3) and (5.4) according to the dimension of the formation and the leader dynamics [7].

A homogeneous system is considered, i.e., all UGVs have the same physical constraints. The UGVs are modeled as a tricycle, then, the linear velocity and steering angle of the followers are constrained by v_{min}, v_{max} and γ_{max}. Hence, they must satisfy:

$$v_{min} \leq |v^c_{T_i}| \leq v_{max} \tag{5.11}$$

$$|\gamma^c_{T_i}| \leq \gamma_{max} \tag{5.12}$$

The dynamic of the target according to the leader dynamic, eq. (5.3) and (5.4), was given in subsection 5.2.1. Therefore, leader constraints such as velocity and steering angle can be defined as functions of the UGV constraints eq. (5.11) and (5.12) and the formation shape.

The steering angle is directly related to the curvature of the UGV $c_c = 1/r_c = tan(\gamma)/l_b$. Therefore, the steering angle constraint eq. (5.12) can be written as curvature constraint $c_{c_{max}}$, moreover, $\omega_L = v_L c_{c_L}$. This representation in function of c_{c_L} is useful for the following computation. To simplify the notation, let us introduce:

$$A(c_{c_L}) = A_{c_L} = (1 - l^c_i c_{c_L})^2 + (h^c_i c_{c_L})^2 \tag{5.13}$$

By using eq. (5.3) and (5.13) in the velocity constraint (5.11), we obtain:

$$v_{min} \leq v_L A^{1/2}_{c_L} \leq v_{max} \tag{5.14}$$

The steering angle's constraint can be written in function of curvature $c_{c_{max}}$. Using eq. (5.3) and (5.4) to compute the curvature of the followers $c_{c_{T_i}} = \omega_{T_i}/v_{T_i}$ with eq. (5.5) and (5.13), we obtain:

$$\left| \frac{c_{c_L}}{A^{1/2}_{c_L}} + \frac{h^c_i \dot{c}_{c_L}}{v_L A^{3/2}_{c_L}} \right| \leq c_{c_{max}} \tag{5.15}$$

where $|\cdot|$ is the absolute value of the expression.

An important term for the constraints is $\dot{c}_{c_L}$. For a tricycle, it is related to the velocity of the steering angle of the leader and it is given by:

$$\dot{c}_{c_L} = \dot{\gamma}_L \sec^2(\gamma_L)/l_b \tag{5.16}$$

where γ_L is the steering angle of the leader and sec is the secant function.

Analyzing eq. (5.14) and (5.15), it is observed that the limits of (5.13) allows to obtain the leader constraints. The first derivative of $A(c_{c_L})$ was computed to obtain the minimum value of (5.13):

$$h^2_m / \left(h^2_m + l^2_m\right) = A_{c_L min} \leq A(c_{c_L}) \leq A(c_{c_L max}) \tag{5.17}$$

where $\mathbf{f}_m = (h_m, l_m)^T$ is the coordinates of the farthest node w.r.t. the instantaneous center of rotation determined by the leader Icc_L. The limits of $\dot{c}_{c_L}$ is given by:

$$\dot{c}_{c_L} \leq \dot{c}_{c_L max} = \dot{\gamma}_{Lmax} \sec^2(\gamma_{Lmax})/l_b \tag{5.18}$$

where γ_{Lmax} is the maximum steering angle of the leader.

Using eq. (5.17) and (5.18) and applying the triangle inequality in eq. (5.14) and (5.15), it is obtained:

$$v_{Lmax} A^{1/2}(c_{c_L max}) < v_{max} \tag{5.19}$$

$$\left|\frac{c_{c_L max}}{A^{1/2}_{c_L min}}\right| + \left|\frac{h_m \dot{c}_{c_L max}}{v_{Lmin} A^{3/2}_{c_L min}}\right| < c_{c_{max}} \tag{5.20}$$

where $v_{Lmin} = v_{min}/A^{1/2}_{c_L min}$.

Finally, leader's constraints v_{Lmax} and $c_{c_{Lmax}}$ which respect to all the followers' physical constraints are obtained while solving numerically two inequalities eq. (5.19) and (5.20).

Nevertheless, these fixed leader's constraints can reduce drastically the dynamic of the leader along its navigation in formation (velocity and steering angle close to their minimum values) and thus the dynamic of the formation. At this aim, the minimum and maximum velocity and maximum curvature (related to maximum steering angle) of the leader are adapted according to its instantaneous values of velocity and curvature. The proposed adaptive constraints (velocity and steering angle using the dynamic of the leader) permit to enhance the trajectory's smoothness and the convergence toward the desired formation and to keep it. The adaptive constraints of the Leader are given by:

$$v_{Lmin} = v_L \tag{5.21}$$

$$v_{Lmax} = v_{max} A^{-1/2}_{c_L} \tag{5.22}$$

$$\left|c_{c_L max}\right| = c_{c_{max}} \left|A^{1/2}_{c_L min}\right| - \left|A^{1/2}_{c_L min}\right| \left|\frac{h_m \dot{c}_{c_L}}{v_{Lmin} A^{3/2}_{c_L}}\right| \tag{5.23}$$

where $A(c_{c_L})$ and $\dot{c}_{c_L}$ are the instantaneous values according to the current c_{c_L} of the leader. These adaptive constraints are obtained from eq. (5.19) and (5.20) to obtain the maximum values (velocity and steering angle) according to the leader curvature (5.13). The adaptation of v_{Lmin} and the instantaneous values of $\dot{c}_{c_L}$ allow us to increase the limits of γ_{Lmax} while decreasing the second term of eq. (5.20). Furthermore, c_{c_L} contributes to increase the limits of v_{Lmax}, e.g., when the leader is moving in straight line, the formation can navigate with v_{max}.

5.3.2/ Constraints in Frenet formation

The Frenet frame allows each UGV to navigate according to a reference trajectory (segment or complete) while keeping a curvilinear formation (cf. Subsection 5.2.2). This formation definition focuses on the tracking of a reference trajectory (or parallel to it) by UGVs to obtain safe navigation of the formation along the road. Nonetheless, the geometric formation shape changes according to the reference path (cf. Section 5.2.3). The leader should start before its movement to communicate its trajectory to other UGVs. Furthermore, the leader can also track a reference trajectory which is known beforehand by all UGVs. The main applications can be founded in urban transport systems [277], autonomous agricultural convoy [234], exploration [302]. Some drawback can be found:

- The formation shape is distorted according to the leader trajectory.
- The leader's trajectory must be known in advance by the followers.

The dynamic of the Frenet formation depends mainly on the reference trajectory (leader's trajectory) eq. (5.6), (5.7) and (5.8). This trajectory is represented in Frenet frame (curvilinear distance and radius of curvature), therefore, the followers' constraints are given by v_{max} and $r_{c_{min}}$. The velocities and radius of curvature of the followers are constrained by

$$v_{min} \leq |v^f_{T_i}| \leq v_{max} \tag{5.24}$$

$$|\omega^f_{T_i}| \leq \omega_{max} \tag{5.25}$$

$$r_{c_{min}} \leq \left| \frac{v^f_{T_i}}{\omega^f_{T_i}} \right| \leq \infty \tag{5.26}$$

Eq. (5.24), (5.25) and (5.26) can be expressed in function of leader's velocities using eq. (5.7) and (5.8) as follows:

$$v_{min} \leq |v_L - l^f_i \omega_L(h^f_i)| \leq v_{max} \tag{5.27}$$

$$0 \leq |\omega_L(h_i)| \leq \omega_{max} \tag{5.28}$$

$$r_{c_{min}} \leq |\frac{v_L - l^f_i \omega_L(h_i)}{\omega_L(h_i)}| \leq \infty \tag{5.29}$$

The leader's linear velocity is obtained from eq. (5.27) and (5.28) as:

$$v_{min} + |l_m \omega_{max}| < |v_L| < \min\{v_{max} - |l_m \omega_{max}|, \omega_{max} r_{c_{min}}\} \tag{5.30}$$

where $\mathbf{f}_m = (h_m, l_m)^T$ is the coordinates of the farthest node and the minimum function $(\min)$ allows to saturate the leader's velocity with the minimum centripetal velocity $(\omega_{max} r_{c_{min}})$. Using eq. (5.8) and (5.29), we obtain:

$$\begin{aligned} r_{c_{min}} &< |r_{c_L}(h_m) - l_m| \\ r_{c_{min}} + |l_m| &< |r_{c_L}(h_m)| \end{aligned} \tag{5.31}$$

It can be noted that the linear velocity v_L and the radius of curvature r_{c_L} of the leader are constrained mainly according to the lateral coordinate of the formation l_m (eq. (5.30) and (5.31)). The linear velocity at h^f_i (curvilinear distance) and $l^f_i = 0$ is equal to the current linear velocity of the leader which allows to keep constant the curvilinear distance (cf. Fig. 5.5(b)).

Finally, solving eq. (5.30) and (5.31), the leader constraints of v_{Lmax} and $r_{c_{Lmin}}$ are obtained.

$$v_{Lmax} = \omega_{max} r_{c_{min}} \tag{5.32}$$

$$r_{c_{Lmin}} = r_{c_{min}} + |l_m| \tag{5.33}$$

These fixed leader's constraints can reduce drastically the dynamic of the leader along navigation in formation (velocity and radius of curvature close to their minimum values) and thus the dynamic of the formation. To address this issue, the adaptive constraints, velocity and radius of curvature using the dynamic of the leader, are proposed. They

permit to improve the convergence toward the desired formation and to keep it. The proposed adaptive constraints of the Leader are given by:

$$r_{c_{Lmin}} = \min\left(|r_{c_L} + l_i^f|\right) \tag{5.34}$$

$$r_{c_{Lmax}} = \max\left(|r_{c_L} + l_i^f|\right) \tag{5.35}$$

$$v_{Lmax} = v_{max} \min\left(r_{c_L}/r_{c_{Lmax}}, 1\right) \tag{5.36}$$

where r_{c_L} is the instantaneous value of radius of the curvature of the leader. These adaptive constraints are obtained from eq. (5.32) and (5.33) to obtain the maximum and minimum values (velocity and radius of curvature, respectively) according to the current radius of curvature of the leader. The minimum function in eq. (5.36) allows to saturate the leader's velocity to the maximum vehicle's velocity v_{max}. In eq. (5.34), the minimum function selection the minimum radius of curvature among all UGVs. Furthermore, r_{c_L} contributes to decrease the limits of v_{Lmax}, e.g., when the leader is moving in straight line and the followers are in a curve, the leader can navigate with v_{max}, however, the followers can not to navigate with this maximum velocity. Therefore, the leader has to limit its velocity to permit the followers to navigate in a curve with suitable velocities.

5.3.3/ SIMULATION ANALYSIS OF FORMATION'S CONSTRAINTS

This subsection shows the application of the proposed dynamic constraints (cf. Section 5.3) to obtain smooth (smooth values of the vehicle commands) navigation in formation (cf. Fig. 5.10 and 5.12).

5.3.3.1/ CARTESIAN FORMATION

The scenario with a group of UGVs navigating in triangular formation $\mathbf{F}_3$ with $v_L = 2\ m/s$ presented in Subsection 5.2.3.1 is used. The leader tracks a reference sinusoidal trajectory and the followers have to track their assigned virtual targets.

This simulation shows the proposed adaptive constraints applied to Cartesian frame. Fig. 5.10 shows the trajectories of the UGVs using the proposed adaptive leader's constraints

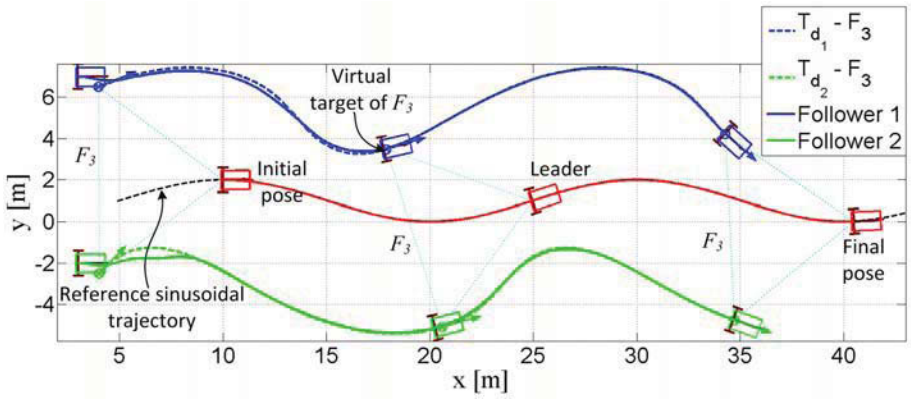
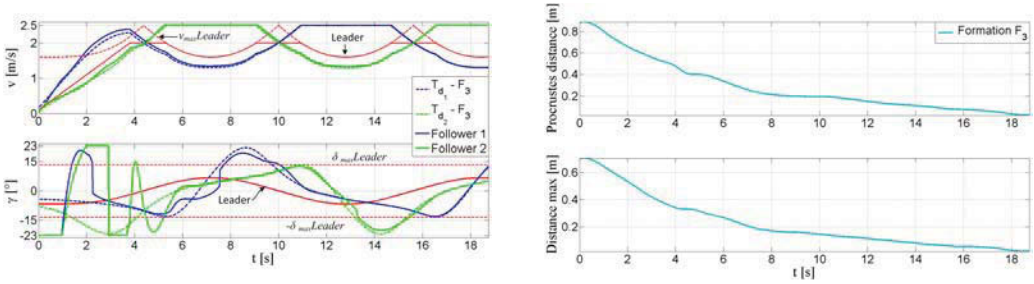


Figure 5.10: Navigation in Cartesian formation $\mathbf{F}_3$ ($v_L max = 2\ m/s$) for a group of $N = 3$ UGVs using the proposed adaptive leader's constraints ($T_{d_i} - F3$ is the i dynamic virtual target w.r.t. $\mathbf{F}_3$ formation).



(a) Commands of the UGVs.

(b) Formation shape $\mathbf{F}_3$ ($v_L max = 2\ m/s$).

Figure 5.11: Simulation results of the navigation with adaptive Leader Constraints (Cartesian formation).

(rely on kinematic vehicles' constraints v_{max} and γ_{max}). The followers' trajectories are smooth and track accurately their assigned targets. Fig. 5.11(a) show the velocity and steering angle of the UGVs using the proposed adaptive constraints. It can be noted that the velocities and steering angle of the followers comply with the kinematic vehicles' constraints (v_{max} and γ_{max}) while the leader satisfies the proposed adaptive constraints $(v_{Lmax}, \gamma_{Lmax})$. Fig. 5.11(b) shows the evolution of the Procrustes distance P_d between the positions of the vertex (cf. Appendix F). It can be observed that the formation using the proposed adaptive leader constraints converges to the desired formation shape $\mathbf{F}_3$.

Table 5.3 shows the values of the $\mathcal{L}_2$ (cf. Appendix F) norm applied to the evolution of P_d and Dn_{max}. This table demonstrates that the use of the proposed adaptive constraints allows to keep formation shape during the navigation even when for fast leader's dynamics (e.g., Table 5.1 shows the navigation without using the adaptive constraints).

	$\mathcal{L}_2(P_d)\ [m]$	$\mathcal{L}_2(Dn_{max})\ [m]$
$v[m/s]$	2.0	2.0
$\mathbf{F}_3$	0.3703	0.2903

Table 5.3: Procrustes distances and maximum distance for different velocities and dimensions

5.3.3.2/ FRENET FORMATION

This simulation shows the proposed adaptive constraints applied to Frenet formation. The used scenario is the same as in Subsection 5.2.3.2, a group of UGVs navigating in triangular formation $\mathbf{F}_3$ w.r.t. leader's trajectory with $v_L = 2\ m/s$.

The trajectories of the UGVs using the proposed adaptive leader's constraints (rely on kinematic vehicles' constraints v_{max} and $r_{c_{min}}$) are shown in Fig. 5.12. It can be noted that the followers perform smooth trajectories and track accurately their assigned targets. The velocity and steering angle of the UGVs using the proposed adaptive constraints for Frenet formation are shown in Fig. 5.13(a). It can be observed that the velocities and steering angle of the followers comply with the kinematic vehicles' constraints (v_{max} and γ_{max}) while the leader satisfies the proposed adaptive constraints $(v_{Lmax}, r_{c_{Lmin}})$. Fig. 5.13(b) shows the evolutions of the distance and orientation errors as in Subsection 5.2.3.2. It can be noted that the errors using the proposed adaptive leader constraints converges to the desired formation shape $\mathbf{F}_3$ for $v_L = 2\ m/s$.

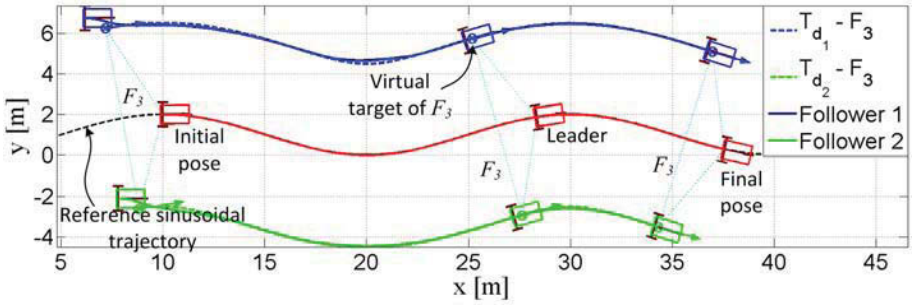
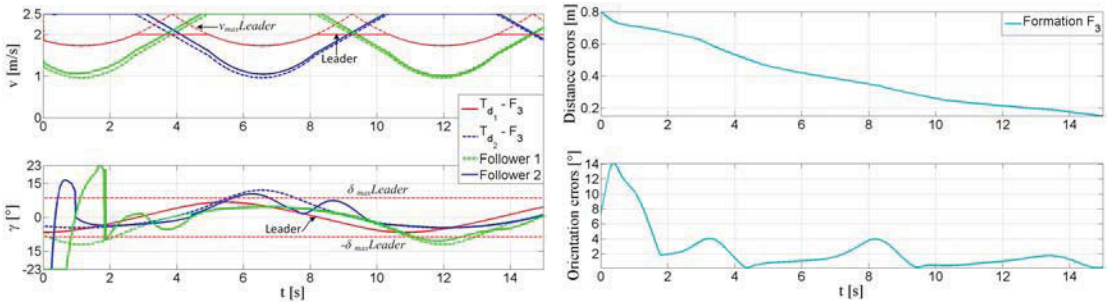


Figure 5.12: Navigation in Frenet formation $\mathbf{F}_3$ ($v_L max = 2\ m/s$) for a group of $N = 3$ UGVs using the proposed adaptive leader's constraints ($T_{d_i} - F3$ is the i dynamic virtual target w.r.t. $\mathbf{F}_3$ formation).



(a) Commands of the UGVs.

(b) Distance and orientation errors $\mathbf{F}_3$ ($v_L max = 2\ m/s$).

Figure 5.13: Simulation results of the navigation with adaptive Leader Constraints (Frenet formation).

Table 5.4 shows the values of the $\mathcal{L}_2$ (cf. Appendix F) norm applied to the evolution of d_{rms} and $e_{\theta_{rms}}$. As in the Cartesian formation case (cf. Subsection 5.3.3.1), this table demonstrates that the use of the proposed adaptive constraints designed for Frenet formation allows to keep formation w.r.t. leader's trajectory during the navigation even when for fast leader's dynamics (e.g., Table 5.2 shows the navigation without using the adaptive constraints).

	$\mathcal{L}_2(d_{rms})$ $[m]$	$\mathcal{L}_2(e_{\theta_{rms}})$ $[°]$
$v[m/s]$	2.0	2.0
$\mathbf{F}_3$	0.4423	3.8867

Table 5.4: Distances and orientation errors for different velocities and dimensions

5.3.4/ SIMULATION AND EXPERIMENTAL VALIDATION OF THE WHOLE CONTROL ARCHITECTURE FOR NAVIGATION IN FORMATION

This subsection presents simulations in COBAYE simulator (cf. Subsection G.2.1) and experiments with urban electric cars (VIPALABs) using the proposed control architecture for navigation in formation (cf. Fig. 5.3 and 5.4). One UGV of the group of VIPALABs is considered as the leader, i.e., the formation will be defined according to its dynamic. The leader of the group of VIPALABs has to reach successively static set of waypoints (cf. Chapter 4) and the followers has to keep the desired formation while avoiding the hinder

obstacles. The Leader's configuration is sent by itself to each follower via WI-FI. At this aim, each follower tracks its assigned dynamic virtual target (cf. Fig. 5.17(a) and 5.17(b)) applying the proposed control law to the multi-robot system.

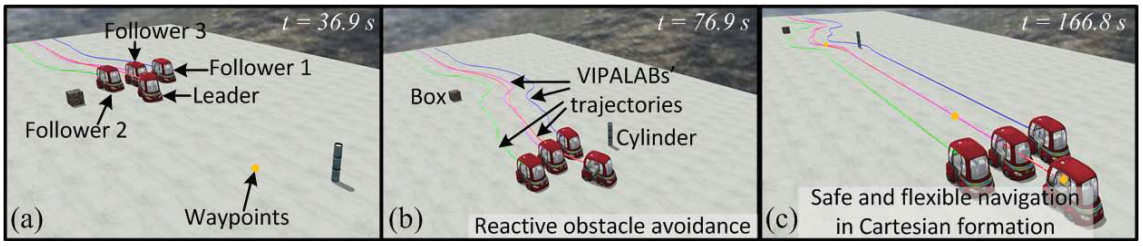
In these simulations and experiments, the used UGVs' parameters, constraints, sensors and controller parameters were described in Section 3.6. Moreover, the communication system by Wi-FI between UGV is stable.

Firstly, this simulation focuses on the performance of the proposed control architecture for navigation in Cartesian formation in **cluttered environment** (cf. Fig. 5.14(a)-(a)). This simulation can be found online[1]

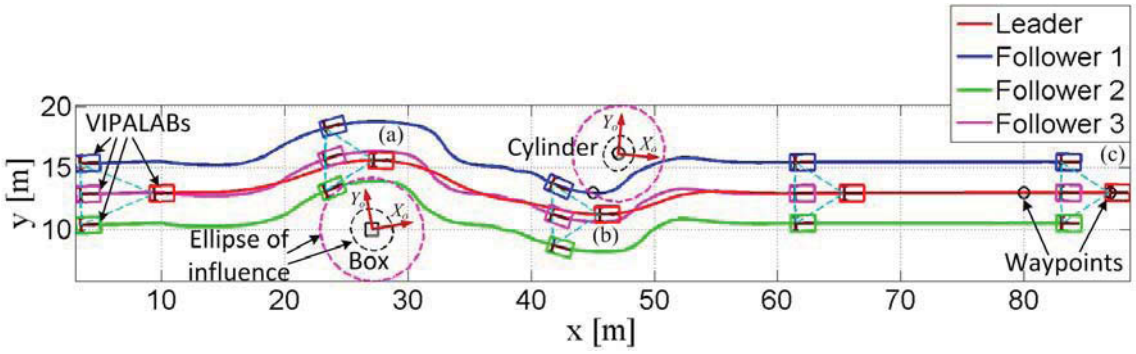
The formation has a triangular shape $\mathbf{F}^c = \left((-4, 2.5)^T, (-4, -2.5)^T, (-4, 0)^T\right)\ m$. The initial positions of the vehicles have an offset $(\Delta x, \Delta y) = (-1.0,\ 0.0)\ m$ from the initial position of their assigned virtual targets. In Fig. 5.14(a), the red line represent leader's trajectory and the big orange points are the set of waypoints. When the leader detects an obstacle with adequate range, then the whole formation will avoid it while keeping the desired shape using the limit-cycle method (limit-cycle is increased by $R_f = 2.5\ m$ to allow a safe navigation (cf. Subsection 3.2)) (cf. Fig. 5.14(a) - (a) and (b)). The followers track their virtual targets to keep the desired formation **F** (cf. Fig. 5.14(a) - (c)).

Fig. 5.14(b) shows the trajectories of the UGVs during the navigation in formation. It can be seen that the leader reaches accurately the successive static waypoints. The followers track accurately their virtual targets even during obstacle avoidance phase. It can be noted that no UGV collides with any obstacle. Fig. 5.15(a) and 5.15(b) show respectively the values of Lyapunov function (cf. eq. (3.32)) for each UGV and the evolution of the Procrustes distance P_d between the positions of the vertex. Some small peaks can be

[1] http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/SimNavCartFor.mp4

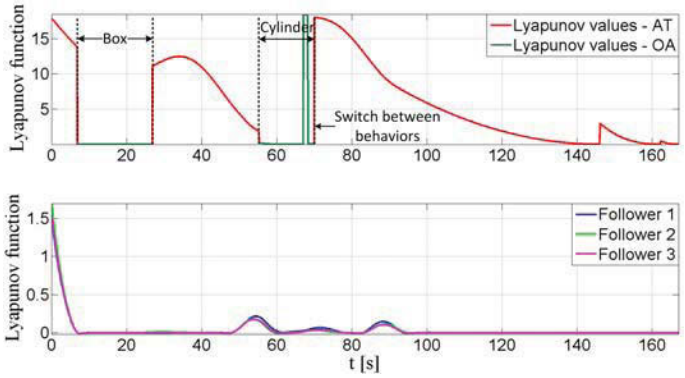


(a) Navigation in Cartesian formation implemented in COBAYE simulator.

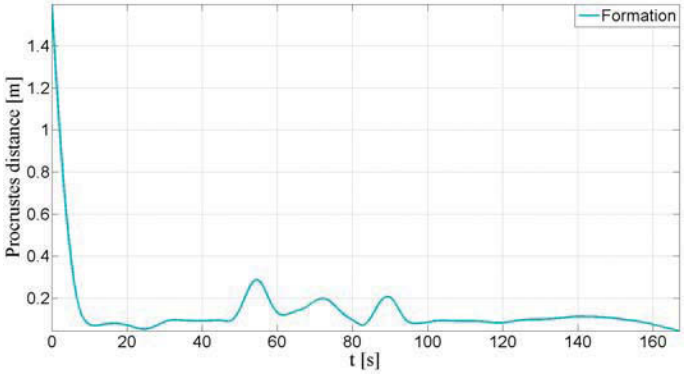


(b) Vehicles' trajectory using the proposed control architecture for navigation in Cartesian formation.

Figure 5.14: Safe navigation in Cartesian formation in cluttered environment.



(a) Lyapunov function of each UGV (AT: Attraction to the Target and OA: Obstacle Avoidance).



(b) Procrustes distance of the Cartesian formation.

Figure 5.15: Validation of the navigation in Cartesian formation.

observed in the followers' Lyapunov function and the Procrustes distance. These peaks are related to the switch between behaviors and the fast dynamic change of the leader (acceleration are increased and a saturation occurs in the followers when the leader curvature is increased). It can be observed in Fig. 5.15(b) that the formation converges to the desired formation shape $\mathbf{F}^c$. Therefore, the proposed Cartesian formation allows safe, flexible and smooth navigation for a group of UGVs while keeping a geometric shape in cluttered environment.

Secondly, this experiment focuses on the performance of the proposed control architecture for navigation in Frenet formation (cf. Fig. 5.4) with three VIPALABs in PAVIN platform (cf. Fig. 5.16). The scenario was carried out to exhibit the safety and flexibility of the proposed navigation in Frenet formation in structured environment, a scenario with the presence of obstacles is presented (cf. Fig. 5.16 (b) and (e)). Two obstacles are placed between the waypoints. This experiment can be found online[2].

The desired formation has a linear shape $\mathbf{F}^f = \left((-5,0)^T,(-10,0)^T\right)$. The initial positions of the followers have an offset $(\Delta x, \Delta y) = (-0.5,\ 0.5)\ m$ from the initial position of their assigned virtual targets.

Some screenshots of the developed Graphical Data Interface for VIPALAB (GDI-VIPA) of the followers are shown in Fig. 5.17(a) and 5.17(b). The white line represent UGV's

[2]http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/NavigationFor.mp4

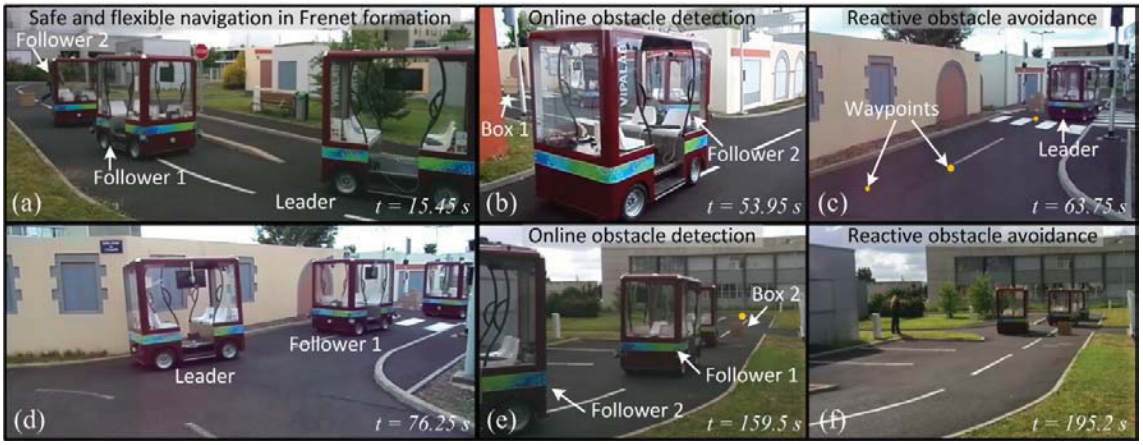
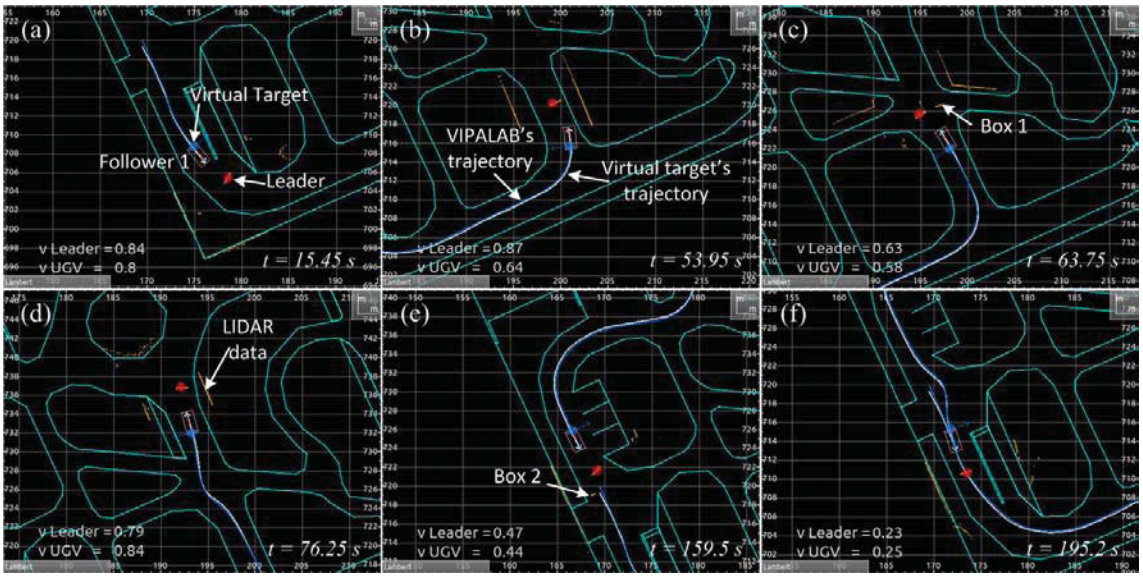


Figure 5.16: Safe navigation in Frenet formation in structured environment.

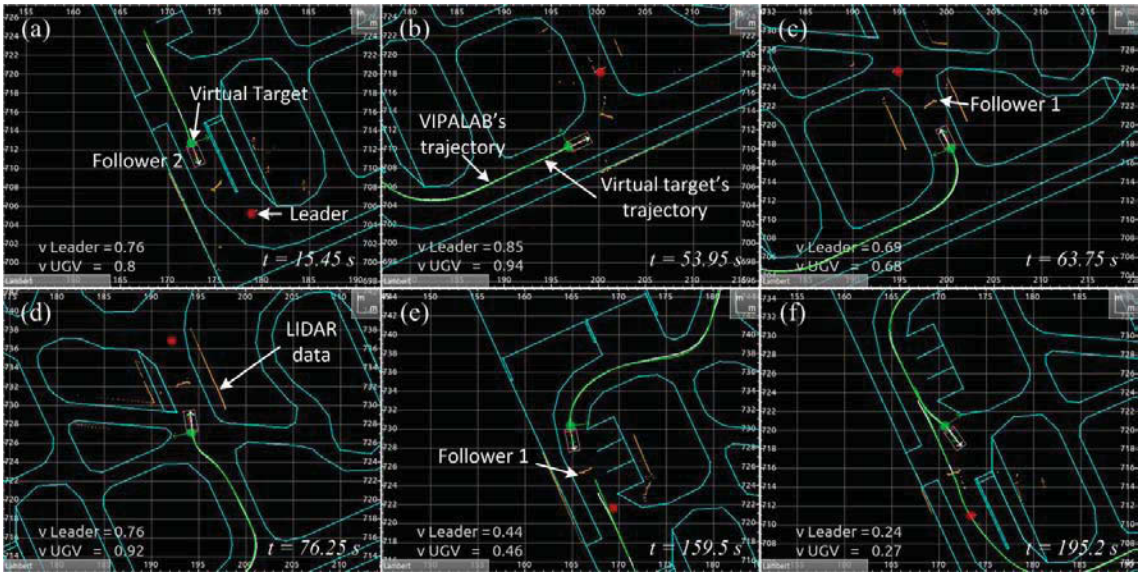
trajectory, the red point is the current leader's pose and the blue and green points are respectively the current virtual target to be tracked by follower 1 and follower 2. The Leader's analysis (navigation through waypoints and reactive obstacle avoidance) was described in Section 4.4.2.

Fig. 5.17(c) shows that the leader reaches accurately the successive static waypoints and the followers track accurately also their dynamic targets (w.r.t. leader). Moreover, the followers' trajectories using the proposed control law is close to the leader's trajectory (cf. Fig. 5.17(c)). The Leader detects the hinder obstacles between the waypoints and it applies the reactive limit-cycle method (cf. Subsection 4.4.2). The Followers avoid also the obstacle since it tracks accurately the leader's trajectory. It can be noted that the followers do not collide with the obstacles and they keep the desired Frenet formation.

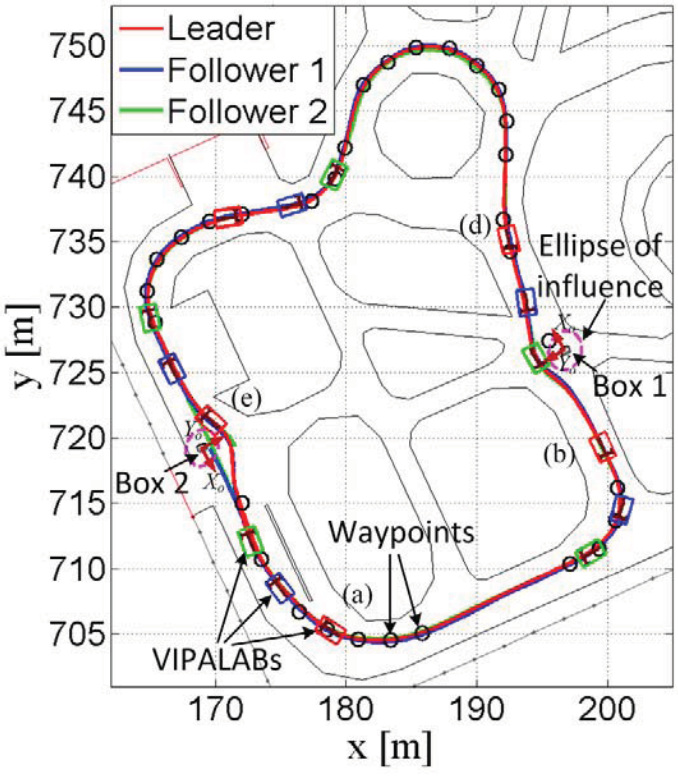


(a)

Figure 5.17: Validation of the navigation in Frenet formation. (a) GDI-VIPA during the navigation of the follower 1 ("v UGV" and "v Leader" are respectively, the follower's and the leader's current velocities).



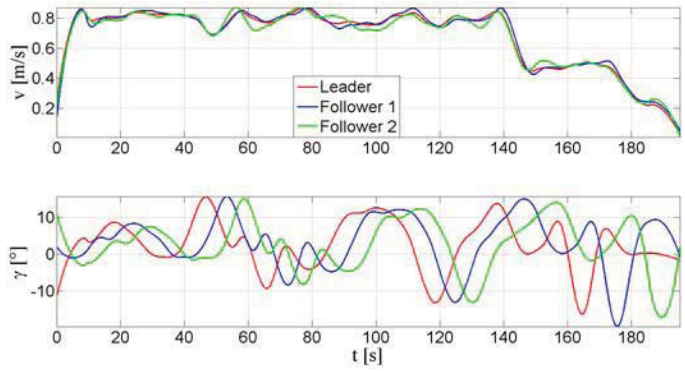
(b)



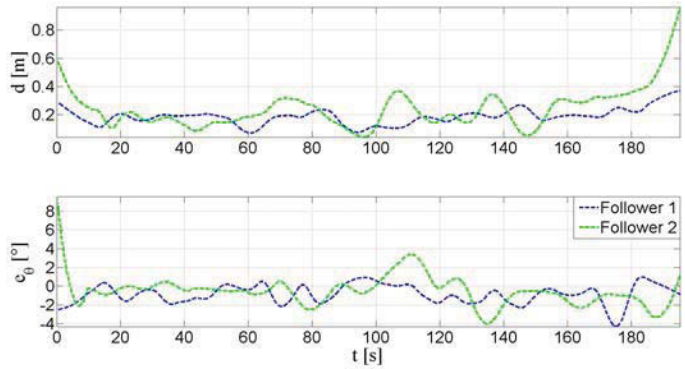
(c)

Figure 5.17: Validation of the navigation in Frenet formation (cont.). (b) GDI-VIPA during the navigation of the follower 2 ("v UGV" and "v Leader" are respectively the follower's and the leader's current velocities) and (c) Vehicles' trajectories using the proposed control architecture for navigation in Frenet formation.

Fig. 5.18(a) shows the velocity and steering angle of the followers. It can be noted that the linear velocity of the followers is very close to the leader's velocity. The steering angle



(a) Commands (velocity and steering angle) of each follower.



(b) Distance and orientation errors of each follower.

Figure 5.18: Experimental results of the navigation in Frenet formation.

of the followers have a delay w.r.t the leader because the followers' poses are $5\ m$ and $10\ m$ (in curvilinear distance related to leader's trajectory) from the current leader's pose.

Fig. 5.18(b) shows the distance and orientation errors of each follower w.r.t. its assigned virtual target. It can be observed that each follower tracks accurately its assigned virtual target. Nevertheless, a delay in the vehicle's communication at the end of the leader trajectory increases the error values. Therefore, the proposed navigation in Frenet formation allows safe, flexible and smooth movements for a group of UGVs in structured environment.

These simulations and experiments showed applications of two presented formation frames, Cartesian and Frenet. The Cartesian frame allows an accurate navigation of a group of UGVs while keeping a rigid shape in cluttered environments, nevertheless this rigid shape is very sensible to this dynamic of the leader and to the dimension of the formation. The Frenet formation frame is more suitable for structured environments since it allows safe movements based on the leader's trajectory, nonetheless, this dependency on the leader's trajectory can generate errors since the followers has to compute its virtual target based on the leader pose which can accumulate localization errors and communication delays.

These results demonstrate the feasibility of the proposed strategy for navigation in formation, in the sequel, the problem of dynamic reconfiguration of the fleet is dealt. A typical example of application of formation reconfiguration is when the formation detects a narrow tight corridor, therefore the formation has to adapt to the corridor dimension to

continue the navigation. The challenge consists in what follows to guarantee the stability and the safety of the multi-vehicle system at the time of transitions between configurations (e.g., line towards square, triangle towards line, etc.) [5].

5.4/ FORMATION RECONFIGURATION

This subsection proposes a new Strategy for Formation Reconfiguration (SFR) based on suitable smooth function between different virtual target configurations (cf. Subsection 5.4.1 and 5.4.2). These strategies allow to obtain a fully reactive architecture in the sense that followers track the instantaneous state (pose and velocity) of their virtual targets (thus, without any use of a reference trajectory or a trajectory planning process). Additionally to the reconfiguration process, one should manage potential collisions between UGVs and allocation of virtual targets to UGVs (cf. Appendix B.3).

The inter-vehicles collisions are avoided during the SFR using a penalty function acting on the vehicle velocities (cf. Appendix B.3). Different algorithms optimizing target assignment can be easily integrated in the proposed control architecture (cf. Fig. 5.2) [316, 345]. In this thesis, the allocation of virtual targets to UGVs is achieved using elementary rules when a formation reconfiguration is required (cf. Section 5.4.3). These rules assign a label H_i of the virtual target T_{d_i} to the UGV_i at the beginning of the experiments. This label is kept by each UGV along the reconfiguration process (cf. Fig. 5.20).

Fig. 5.19 shows the *Formation parameters* block of the proposed control architecture (cf. Fig. 5.2) for navigation in formation of a group of UGVs with dynamic reconfiguration according to the environment context. The UGV_L (Leader) determines the formation shape since its configuration is used to compute the virtual targets T_{d_i} of the followers (by *Communication* block in Fig. 5.2). The coordinates of the formation **F** are *a priori* defined by the designer. The UGVs use their range sensor to detect any unforeseen obstacle (cf. Fig. 5.19). The formation reconfiguration strategy is activated when any hinder obstacle is detected. This strategy takes into account the current environmental state and the current T_{d_i} pose to obtain a new formation shape (cf. Fig. 5.19). Moreover, in the case where the obstacle is close enough to any UGV, the proposed control architecture embedded in each UGV allows to have the capability of reactive obstacle avoidance (cf. Subsection

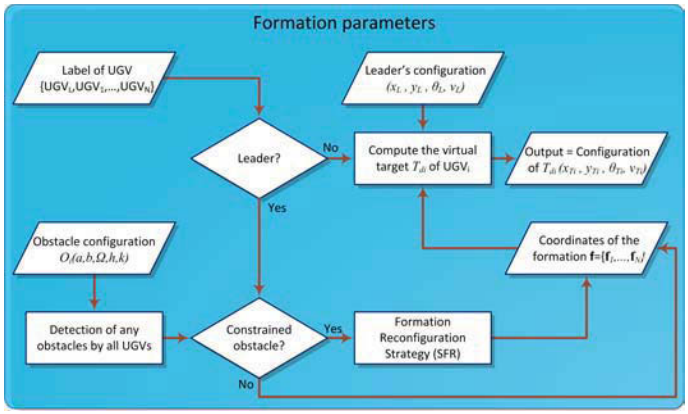


Figure 5.19: Flowchart of the *Formation parameters* block for navigation in formation with dynamic reconfiguration.

3.2).

In this section, two strategies for formation reconfiguration are proposed. The first is a reactive strategy based on smooth switches between initial and final positions of targets, each UGV has to avoid the collision between UGVs. The second is a centralized extension of the first strategy while considering the distance between targets to change smoothly the targets from their initial to final poses.

5.4.1/ Strategy for Formation Reconfiguration based on Smooth Switches (SFR-S)

This reactive strategy for formation reconfiguration is based on suitable smooth switching of the virtual target configurations [5]. The new virtual targets defined for the new formation shape must be ahead to the UGVs to guarantee the stability of the overall system (the vehicle must not go back to reach the new virtual target). If this condition is not satisfied then the former formation will be adapted by increasing smoothly and contentiously the longitudinal coordinates h_i (cf. Fig. 5.3 and 5.4) until that all UGVs will be positioned in the right configuration (5.38). The error between the coordinates of the initial and new formations $\mathbf{e}_{f_i}(e_{h_i}, e_{l_i})$ is defined as:

$$\mathbf{e}^i_{f_i} = \mathbf{f}^n_i - \mathbf{f}^i_i \tag{5.37}$$

where $\mathbf{f}^i_i(h^i_i, l^i_i)$ and $\mathbf{f}^n_i(h^n_i, l^n_i)$ are respectively the coordinates of the initial and new desired formations (cf. Fig. 5.3 and 5.20).

The reconfiguration process between the different formation shapes is given by:

$$\mathbf{f}_i = \begin{cases} h_i = h^n_i - e^i_{h_i} e^{-k_r(t-t_r)}, & l_i = l^n_i; \quad \text{if } e_{h_i} < 0 \\ h_i = h^n_i, & l_i = l^n_i; \quad \text{if } e_{h_i} \geq 0 \end{cases} \tag{5.38}$$

where $\mathbf{f}_i(h_i, l_i)$ are the coordinates of the current virtual target T_{d_i} to be tracked by the follower UGV$_i$. e_{h_i} is the longitudinal coordinate of $\mathbf{e}_{f_i}$ that allows to detect if the virtual target is ahead to its respective follower ($e_{h_i} \geq 0$). The adaptation function when $e_{h_i} < 0$ (virtual target behind to follower$_i$) is set as proportional to the error between formation shapes, where k_r is a real positive constant designed according to the dynamic of the leader and $t_r > 0$ is the initial time for the reconfiguration process. The exponential function guarantee that the longitudinal coordinate converges to the new value h^n_i.

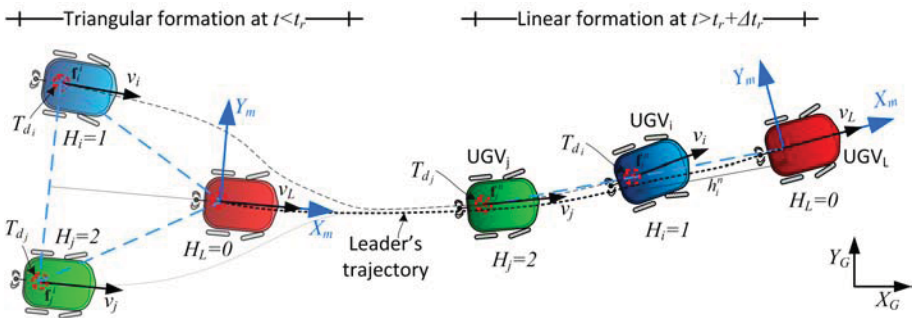


Figure 5.20: Formation reconfiguration between, for instance, triangular and linear formation shapes.

5.4.2/ STRATEGY FOR FORMATION RECONFIGURATION BASED ON INTER-TARGET DISTANCES (SFR-D)

The proposed strategy for formation reconfiguration is an extension of the SFR-S using a continuous functions instead of smooth switches. It is based on the Procrustes distance (cf. Appendix F) which uses the distance errors between the vertex of the initial and new formation (cf. Fig. 5.21). As described in the last subsection, the formation error of a vertex $\mathbf{e}_{f_i}$ is defined by eq. (5.37).

The proposed reconfiguration strategy analyzes the derivative of $\mathbf{e}_{f_i}$ to guarantee the convergence to the new formation shape and smooth trajectories of the virtual target during the reconfiguration process. Moreover, the proposed function has the following form:

$$\dot{\mathbf{e}}_{f_i} = g\left(\mathbf{e}_{f_1}, \ldots, \mathbf{e}_{f_i}, \ldots, \mathbf{e}_{f_N}\right) \tag{5.39}$$

where $\dot{\mathbf{e}}_{f_i}$ takes into account formation errors of all N virtual targets of the formation. It allows to manage the minimum inter-target distance to avoid collision between the followers. The proposed function $\dot{\mathbf{e}}_f$ of the whole formation is designed as a linear system:

$$\dot{\mathbf{e}}_f = \mathbf{A}\mathbf{e}_f \tag{5.40}$$

where $\mathbf{e}_f = [\mathbf{e}_{f_1}, \ldots, \mathbf{e}_{f_N}]^T$ and $\mathbf{A}$ is a symmetric negative definite matrix, it has the following form:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_{12} & \ldots & a_{1N} \\ -a_{12} & a_2 & \ldots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{1N} & -a_{2N} & \ldots & a_N \end{bmatrix} \tag{5.41}$$

where a_i with $i = 1, \ldots, N$ is related to the speed of convergence of the formation parameter $\mathbf{e}_{f_i}$, and a_{ij} with $i \neq j$ is related to the inter-target distances between T_{d_i} and T_{d_j}. The values of the matrix must be designed while taking into account the kinematic constraints of the UGVs to allows each follower to track accurately its assigned target.

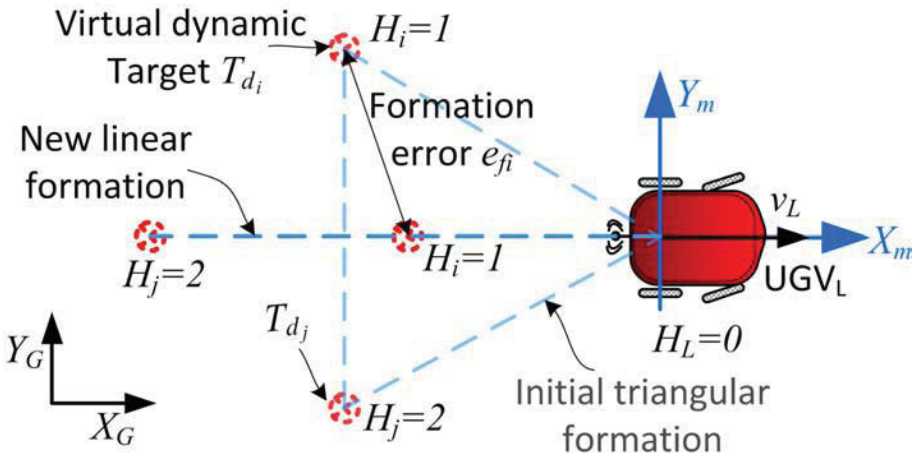


Figure 5.21: Formation reconfiguration between, for instance, triangular and linear formation shapes based on inter-target distances.

The stability of formation error system can be straightforwards proved using Lyapunov analysis (cf. Appendix E.2) with the following Lyapunov candidate function:

$$V = \frac{1}{2}\mathbf{e}_f^T\mathbf{e}_f \tag{5.42}$$

V is a positive-definite function. To guarantee the stability of the system, $\dot{V}$ must be negative-definite. By taking the derivative of eq. 5.42 and using (5.40), $\dot{V}$ can be written:

$$\begin{aligned}\dot{V} =& \mathbf{e}_f^T\dot{\mathbf{e}}_f \\ & \mathbf{e}_f^T\mathbf{A}\mathbf{e}_f\end{aligned} \tag{5.43}$$

Since **A** is negative-definite function, then $\dot{V} < 0$ and the formation error system has an asymptotically convergence.

To avoid the collision between targets, the elements of matrix **A** must be designed according to the desired minimum distance between targets d_{Tmin}. For simplicity, the case of two targets is analyzed. The inter-target error $\mathbf{e}_{f12}$ is defined as:

$$\begin{aligned}\mathbf{e}_{f12} =& \mathbf{f}_1 - \mathbf{f}_2 \\ =& -\mathbf{f}_1^n + \mathbf{f}_1 + \mathbf{f}_2^n - \mathbf{f}_2 + \mathbf{f}_1^n - \mathbf{f}_2^n \\ =& -\mathbf{e}_{f_1} + \mathbf{e}_{f_2} + \mathbf{e}_{f12}^n\end{aligned} \tag{5.44}$$

Furthermore, the inter-target distance can be computed by:

$$d_T^2 = \mathbf{e}_{f12}^T\mathbf{e}_{f12} \tag{5.45}$$

Taking the derivative of eq. (5.45) to obtain its minimum value:

$$\begin{aligned}\frac{d(d_T^2)}{dt} =& 0 \\ \frac{d(\mathbf{e}_{f12}^T\mathbf{e}_{f12})}{dt} =& \\ 2\mathbf{e}_{f12}^T\dot{\mathbf{e}}_{f12} =& 0\end{aligned} \tag{5.46}$$

Eq. (5.46) can be expressed using the derivative of eq. (5.44) with eq. (5.40) as follows:

$$\begin{aligned}\mathbf{e}_{f12}^T\dot{\mathbf{e}}_{f12} =& 0 \\ \mathbf{e}_{f12}^T\left[\dot{\mathbf{e}}_{f_1} - \dot{\mathbf{e}}_{f_2}\right] =& \\ \mathbf{e}_{f12}^T\left[(a_1 + a_{12})\mathbf{e}_{f_1} + (a_{12} - a_2)\mathbf{e}_{f_2}\right] =& 0\end{aligned} \tag{5.47}$$

Using eq. (5.44) in eq. (5.47), it is obtained:

$$\begin{aligned}\mathbf{e}_{f12}^T\left[(a_1 + a_{12})\mathbf{e}_{f_1} + (a_{12} - a_2)\mathbf{e}_{f_2}\right] =& 0 \\ \mathbf{e}_{f12}^T\left[(a_1 + a_{12})\mathbf{e}_{f_1} + (a_{12} - a_2)(\mathbf{e}_{f_1} + \mathbf{e}_{f12} - \mathbf{e}_{f12}^n)\right] =& \\ \mathbf{e}_{f12}^T\left[(a_1 - a_2 + 2a_{12})\mathbf{e}_{f_1} + (a_{12} - a_2)\mathbf{e}_{f12} - (a_{12} - a_2)\mathbf{e}_{f12}^n)\right] =& 0\end{aligned} \tag{5.48}$$

Defining $m_{12} = (a_1 - a_2 + 2a_{12})/(a_2 - a_{12})$ and replacing in eq. 5.48, it is obtained:

$$\begin{aligned}\mathbf{e}_{f12}^T\left[(a_1 - a_2 + 2a_{12})\mathbf{e}_{f_1} + (a_{12} - a_2)\mathbf{e}_{f12} - (a_{12} - a_2)\mathbf{e}_{f12}^n)\right] =& 0 \\ \mathbf{e}_{f12}^T\left[m_{12}\mathbf{e}_{f_1} + \mathbf{e}_{f12}^n)\right] - \mathbf{e}_{f12}^T\mathbf{e}_{f12} =& 0\end{aligned} \tag{5.49}$$

Using the minimum distance between targets d_{Tmin} and eq. (5.45), then eq. (5.49) can be expressed as an inequality:

$$\mathbf{e}_{f12}^T \left[m_{12}\mathbf{e}_{f_1} + \mathbf{e}_{f12}^n) \right] = \mathbf{e}_{f12}^T \mathbf{e}_{f12} \geq \mathbf{e}_{f12min}^T \mathbf{e}_{f12min}$$

$$\mathbf{e}_{f12}^T \left[m_{12}\mathbf{e}_{f_1} + \mathbf{e}_{f12}^n) \right] \geq d_{Tmin}^2 \tag{5.50}$$

$$\tag{5.51}$$

Analyzing the left side of eq. (5.50), the value of m_{12} has to be chosen to produce $\left| m_{12}\mathbf{e}_{f_1} + \mathbf{e}_{f12}^n) \right| \geq d_{Tmin}$ to satisfy the minimum inter-target distance. Furthermore, the values of a_1 and a_2 have to be chosen according to maximum UGV's velocity $a_i \ll v_{max}$ since $\dot{\mathbf{e}}_{f_i}$ is related to the variation $\mathbf{f}_i = (h_i, l_i)$ (cf. eq. (5.37)) which are used for the computation of the virtual target's velocity (cf. eq. (5.3) and (5.7)). The extension for more than two targets follows the same process while taking into account the errors between other targets, the value of m_{ij} determines significantly the minimum distance between the targets T_{d_i} and T_{d_j}.

5.4.3/ VALIDATION OF THE DYNAMIC RECONFIGURATION OF FORMATION

This section shows the navigation of a group of UGVs in a cluttered environment using the proposed strategies for reconfiguration (SFR-S and SFR-D) in simulations and experiments.

In these simulations and experiments, the used UGVs' parameters, constraints, sensors, communication and controller parameters were described in Section 3.6 and 5.3.4. The initial positions of the vehicles have an offset $(\Delta x, \Delta y) = (1,\ 0.5)\ m$ from the initial positions of their assigned virtual targets.

5.4.3.1/ SFR-S RESULTS

The simulation given in Fig. 5.22 to 5.23 focuses on the proposed reconfiguration method based on suitable switches between two Cartesian formation shapes (triangular and linear shapes) while navigating in a cluttered environment. This simulation can be found online[3].

The initial formation coordinates are defined by $\mathbf{F}^i = (\mathbf{f}_1^i, \mathbf{f}_2^i)$, with $\mathbf{f}_1^i = (-4, -2)^T\ m$ and $\mathbf{f}_2^i = (-4, 2)^T\ m$ (triangular shape). Therefore, the group of UGVs must keep the formation while moving in a cluttered environment. A static target is defined in the environment, the leader (and thus the formation) must go toward it while avoiding hinder obstacles. The new targeted formation is defined as straight line with the following coordinates $\mathbf{F}^n = (\mathbf{f}_1^n, \mathbf{f}_2^n)$, with $\mathbf{f}_1^n = (-6, 0)^T\ m$ and $\mathbf{f}_2^n = (-3, 0)^T\ m$. The value of k_r (cf. eq. 5.38) is set to 1 for both follower in the reconfiguration process (when a hinder obstacle is detected).

At the beginning of the simulation (cf. Fig. 5.22), the navigation of the group of UGVs is in triangular formation $\mathbf{F}^i$. When the leader detects an obstacle with adequate range to allow the formation reconfiguration, then the leader avoids the obstacle using the limit-cycle method (limit-cycle is increased by $R_f = 2\ m$ to allow a safe navigation (cf. Subsection 3.2)) and sends the new desired formation $\mathbf{F}^n$ to the other UGVs (followers) to modify the configuration of the formation. The formation returns to triangular shape $\mathbf{F}$, when

[3] http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/FormationReconfiguration.mp4

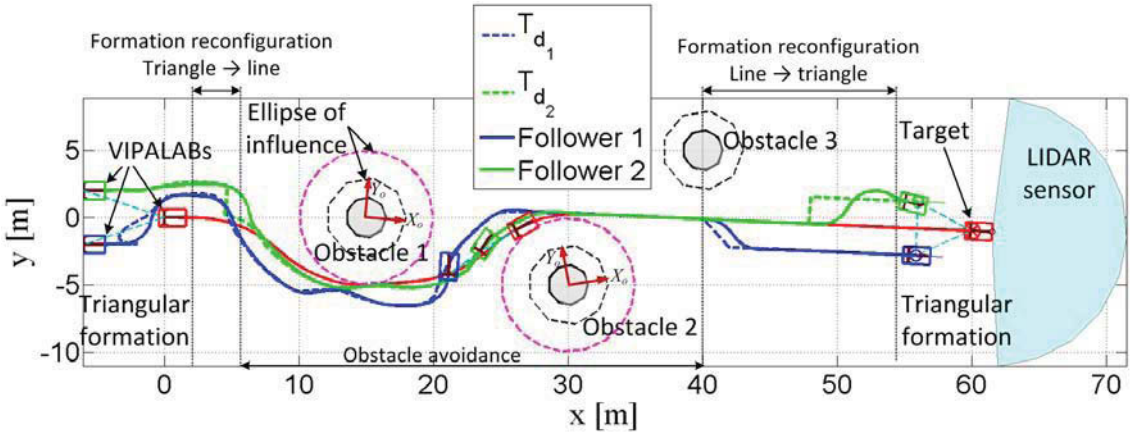
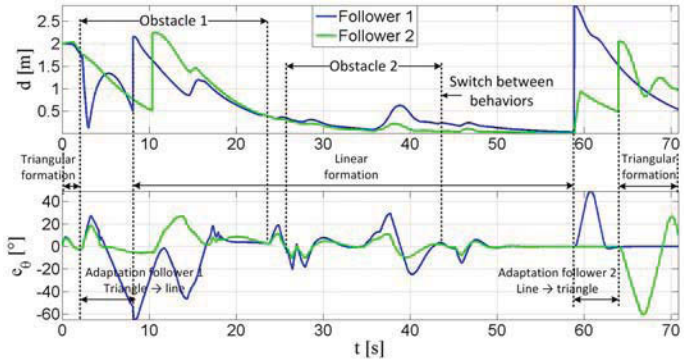
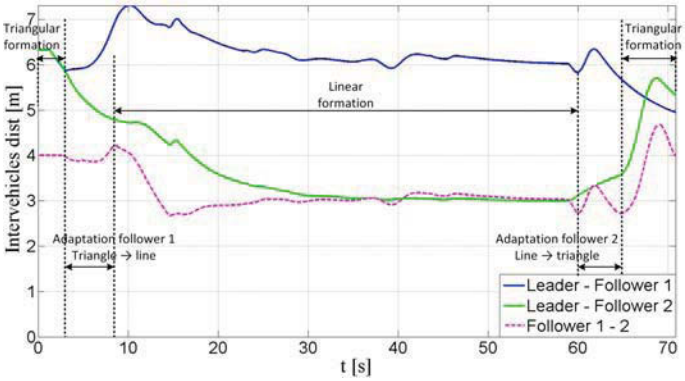


Figure 5.22: Navigation with reconfiguration (SFR-S) in Cartesian formation for a group of $N = 3$ UGVs.

the leader does not detect obstacles that can hinder the other UGVs' movement and the farthest follower left behind the avoided obstacle. The adaptation phase allows to have the virtual target always ahead to the followers to obtain a suitable adaptive formation reconfiguration (cf. Fig. 5.23(a) and 5.23(c)).

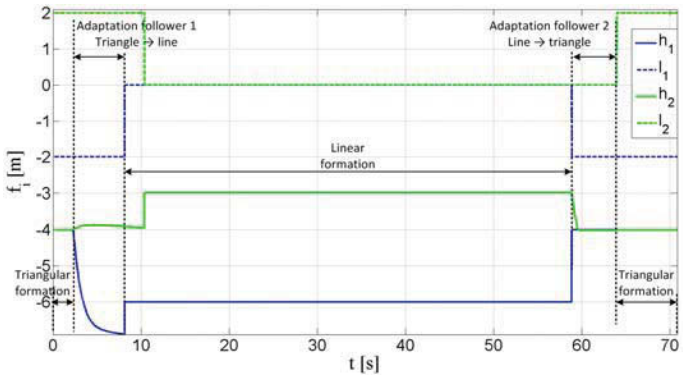


(a) Distance and orientation errors of the UGVs w.r.t. their virtual targets.



(b) Distance among the UGVs.

Figure 5.23: Simulation results of the navigation with reconfiguration (SFR-S).



(c) Progress of the set-point definition $\mathbf{f}_i$ according to the proposed SFR-S.

Figure 5.23: Simulation results of the navigation with reconfiguration (SFR-S) (cont.).

It can be noted in Fig. 5.22 that the vehicles trajectories are smooth along the navigation and there is not neither collisions with the obstacles nor inter-vehicle collisions. Fig. 5.23(a) shows the values of errors d and e_θ between each UGV and its virtual target. At first reconfiguration, it can be observed that the follower 1 wait until its assigned virtual target is ahead (cf. Subsection 5.4.1). Moreover, some small peaks can be observed, they are related to the fast dynamic change of the leader (the dynamic of the formation increased and the saturation occurs in the followers when the leader curvature is increased). Fig. 5.23(b) shows the distance between each UGV of the formation. This last figure shows clearly the non-collision between the vehicles, i.e., $d_{ij} > R_{int_{12}}$ (cf. Appendix B.3). The figures show some peaks which are related to the adaptation and reconfiguration phase between formations. Fig. 5.23(c) shows the evolution of the formation coordinates (h_i, l_i) (virtual target positions). It can be observed that adaptation phase of h_i when the follower is always ahead of its new assigned virtual target (5.38). This attests on the efficiency of the strategy for formation reconfiguration.

The proposed control architecture allows thus to adapt the formation according to the environment context.

5.4.3.2/ SFR-D RESULTS

The objective of this experiment is to validate the proposed control architecture for navigation in formation and the proposed strategy for formation reconfiguration based on inter-target distance between two Frenet formation configurations (triangular and linear shapes) while navigating in a cluttered environment. This experiment has been done while using three VIPALABs (cf. Fig. 5.24). This experiment can be found online[4].

The initial formation coordinates are defined by $\mathbf{F}^i = \left(\mathbf{f}_1^i, \mathbf{f}_2^i\right)$, with $\mathbf{f}_1^i = (-5, -3)^T\ m$ and $\mathbf{f}_2^i = (-5, 3)^T\ m$ (triangular shape). Therefore, the group of UGVs must keep the Frenet formation while moving in a cluttered environment. A set of waypoints is defined in the environment, the leader (and thus the formation) must go toward them while avoiding hinder obstacles. The new targeted formation is defined as straight line with the following coordinates $\mathbf{F}^n = (\mathbf{f}_1^n, \mathbf{f}_2^n)$, with $\mathbf{f}_1^n = (-5, 0)^T\ m$ and $\mathbf{f}_2^n = (-10, 0)^T\ m$. The value of the matrix

[4] http://maccs.univ-bpclermont.fr/uploads/Profiles/VilcaJM/NavigationDynFor.mp4

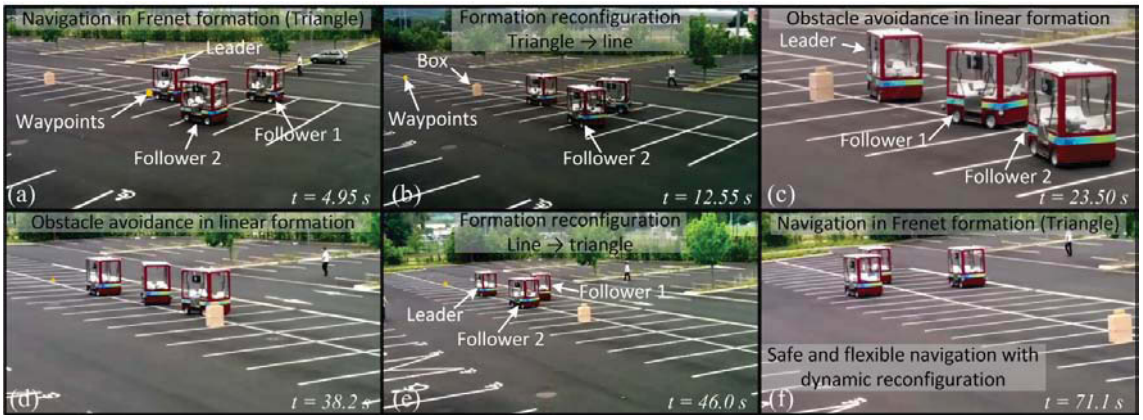
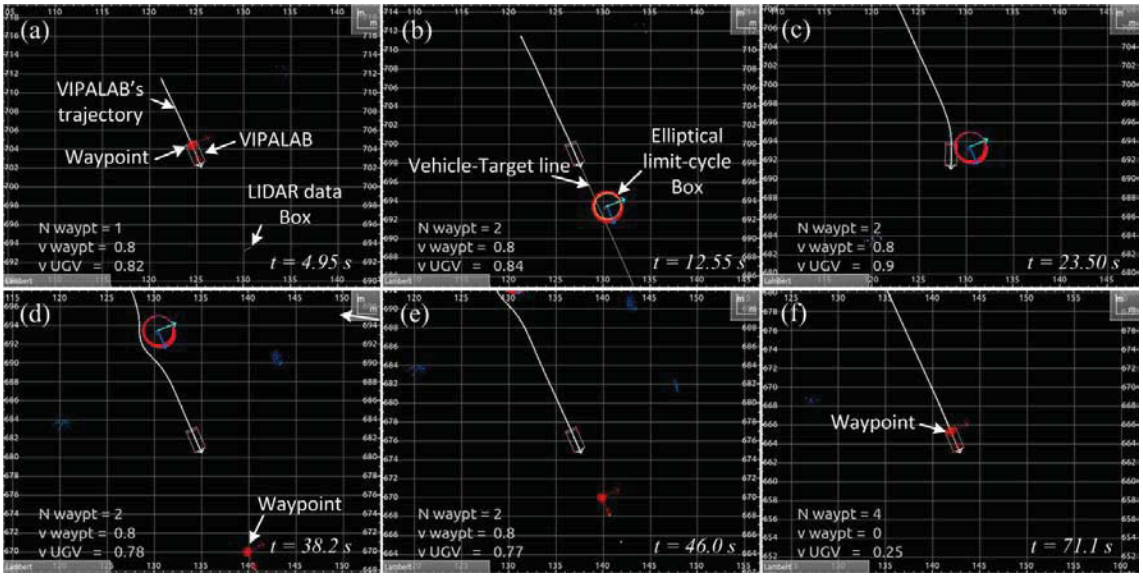


Figure 5.24: Navigation with reconfiguration (SFR-D) in Frenet formation for a group of $N = 3$ UGVs.

A for the formation reconfiguration with $m_{12} = 1.34$ ($d_{Tmin} = 2.0\ m$) is given by:

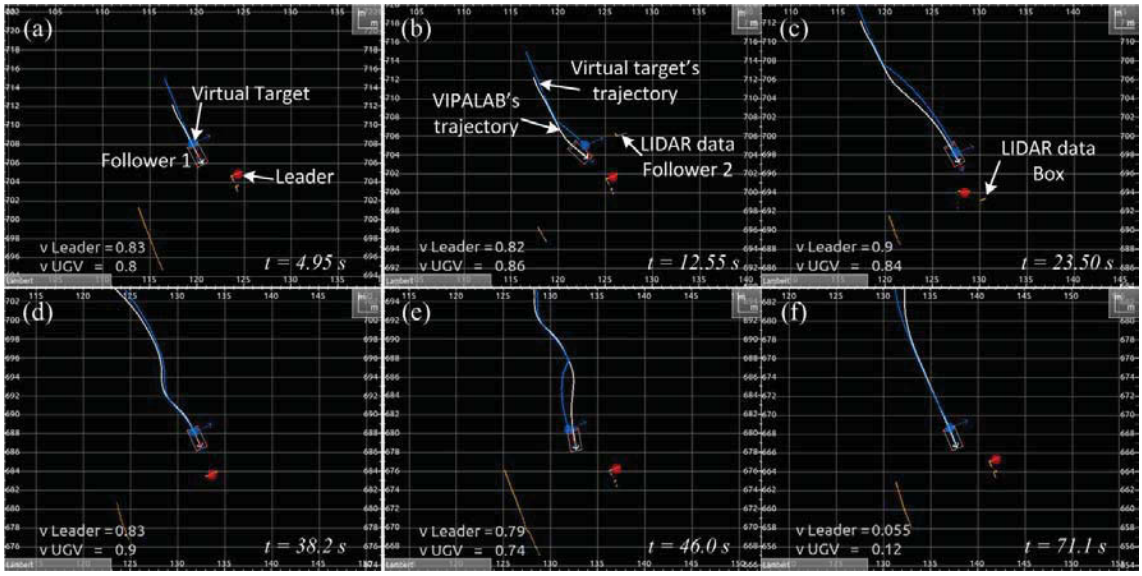
$$\mathbf{A} = \begin{bmatrix} -0.114 & 0.018 \\ -0.018 & -0.143 \end{bmatrix} \tag{5.52}$$

Figure 5.24 shows the sequence of the multi-robot evolution, from the beginning of navigation with initial triangular formation $\mathbf{F}^i$ to linear one $\mathbf{F}^n$, when the leader detects an obstacle (with adequate range to allow the formation reconfiguration), and once the last follower detects the end of the obstacle, the formation return to triangular formation with smooth changes in the virtual targets to always stay ahead to the followers (cf. Fig. 5.26(b) and 5.26(d)).

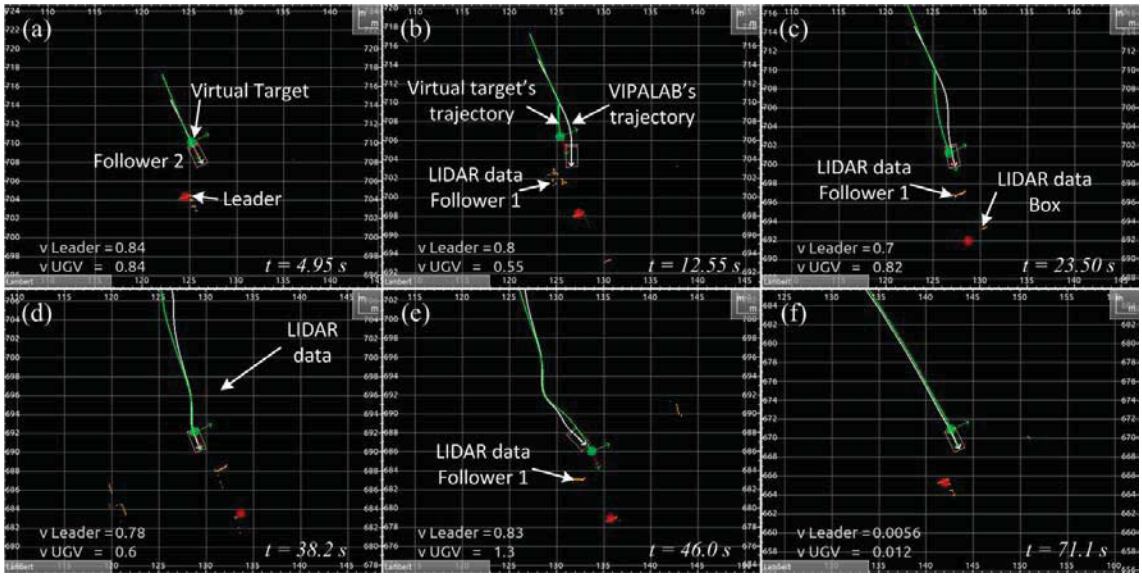


(a)

Figure 5.25: Validation of the navigation with reconfiguration (SFR-D) in Frenet formation for a group of $N = 3$ UGVs. (a) GDI-VIPA during the Leader's navigation ("v UGV" and "v waypoint" are respectively the current velocities of the UGV and the "N waypoint").



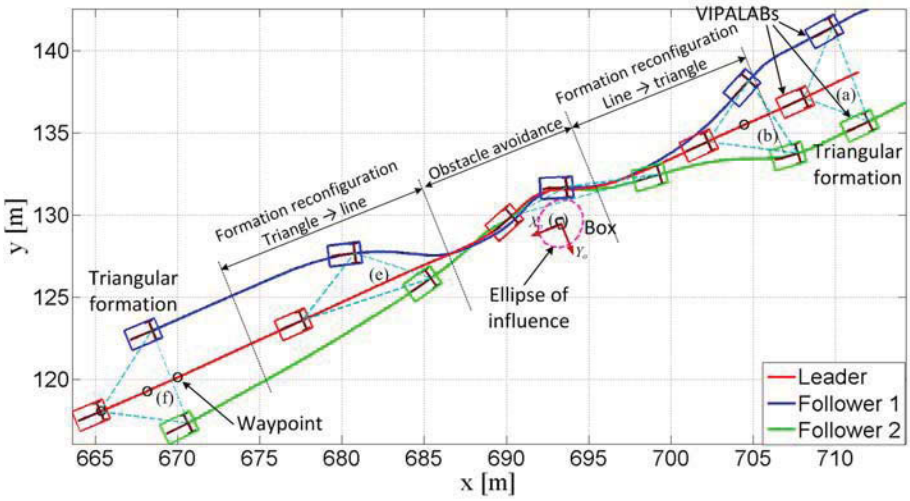
(b)



(c)

Figure 5.25: Validation of the navigation with reconfiguration (SFR-D) in Frenet formation for a group of $N = 3$ UGVs (cont.). (b) GDI-VIPA during the navigation of the follower 1 and (c) follower 2 ("v UGV" and "v Leader" are respectively the follower's and the leader's current velocities).

Some screenshots of the developed Graphical Data Interface for VIPALAB (GDI-VIPA) of the leader and followers are shown in Fig. 5.25(a), 5.25(b) and 5.25(c). In the leader's GDI-VIPA, the white line represents UGV's trajectory, the orange points are the set of waypoints and the big red point is the current assigned waypoint. It can be noted that the online detection of the box using the LIDAR sensor (cf. Section 3.4) and the reactive obstacle avoidance (based on elliptical limit-cycle (cf. Section 3.2)) performed by the UGV (cf. Fig. 5.25(a) (b), (c) and (d)). In the follower's GDI-VIPA, the red point is the current leader's pose and the blue and green points are respectively the current virtual

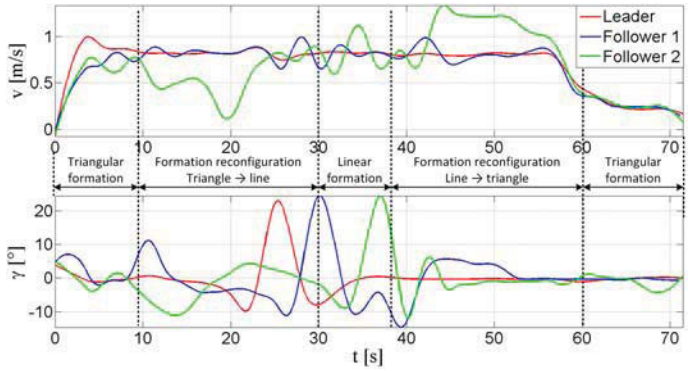


(d)

Figure 5.25: Validation of the navigation with reconfiguration (SFR-D) in Frenet formation for a group of $N = 3$ UGVs (cont.). (d) Vehicles' trajectories using the proposed SFR-D.

target to be tracked by follower 1 and 2. The leader's analysis (navigation through waypoints and reactive obstacle avoidance) was described in Subsection 4.4.2. Therefore, the followers track their virtual targets to keep the desired Frenet formation **F** even during the reconfiguration phase (cf. Fig. 5.24).

Figure 5.25(d) shows safe and smooth trajectories of the group of VIPALABs. It can be observed that the vehicles keep the desired formation while avoiding obstacles. Fig. 5.26(a) shows the velocity and steering angle of the vehicles. The reconfiguration strategy was designed to reduce the peaks of the control commands of each UGV when the transitions between the formation occur. Fig. 5.26(b) shows the Lyapunov function evolution (cf. eq. 3.32) of each UGV which highlight that each vehicle is stable and converges to its assigned virtual dynamic target. At reconfiguration time, it can be observed some small peaks that are related to the switch between formation shapes which cause an abrupt jump of set-points. Fig. 5.26(c) shows the distance between each UGV of the formation. This last figure shows clearly the non-collision between the vehicles in the formation. Fig.

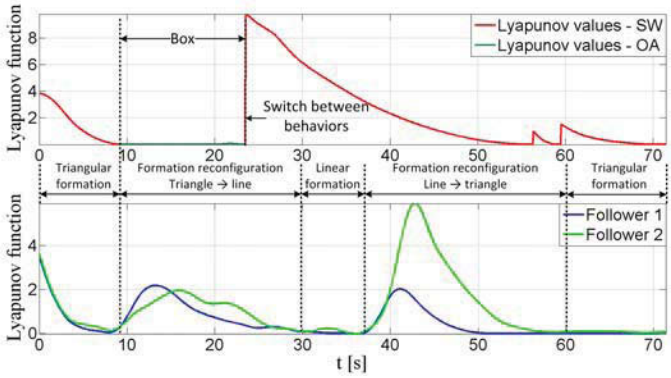


(a) Commands (velocity and steering angle) of each follower.

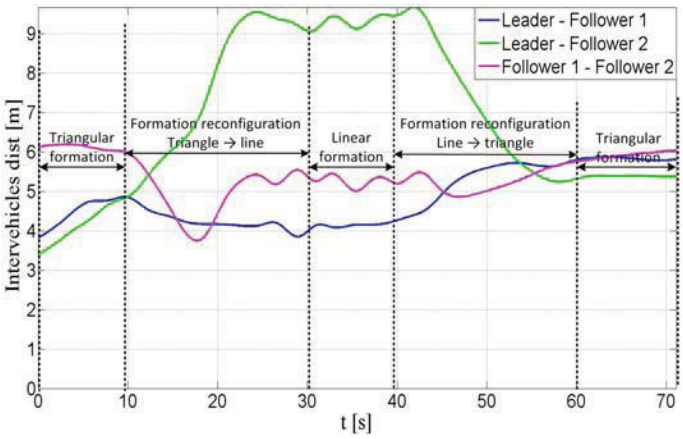
Figure 5.26: Experimental results of the navigation with reconfiguration (SFR-D).

5.26(d) shows the evolution of the formation coordinates (h_i, l_i) (virtual target positions) w.r.t. the leader. It can be observed smooth evolution of the formation coordinates (5.40) which attest on the efficiency of the strategy for formation reconfiguration.

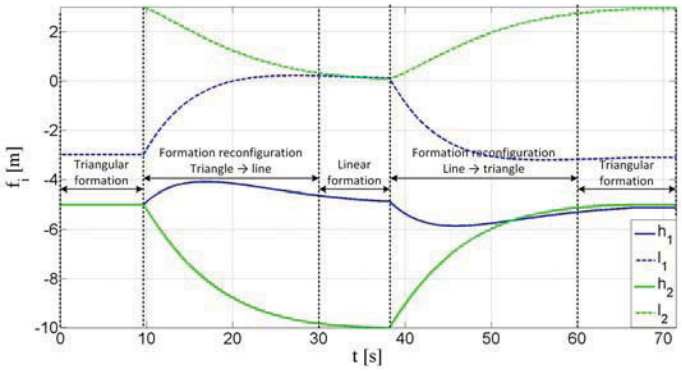
Therefore, a smooth, flexible and safe trajectories for the multi-robot navigation in formation were obtained. The proposed control architecture allows also to adapt the formation configuration according to the environment context.



(b) Lyapunov function of the UGVs w.r.t. their virtual targets (SW: Sequential Waypoints and OA: Obstacle Avoidance.



(c) Distance among the UGVs.



(d) Progress of the set-point definition $\mathbf{f}_i$ according to the proposed SFR-D.

Figure 5.26: Experimental results of the navigation with reconfiguration (SFR-D) (cont.).

5.5/ CONCLUSION

This chapter presented a global control architecture to cope with the navigation in formation for a group of UGV in cluttered and structured environments. Two frame definitions (Cartesian and Frenet) for the formation, its advantages and drawbacks were analyzed and integrated to the proposed global control architecture. In the Cartesian formation, the leader's reference path is not taken into account such as in Frenet formation, only its current pose and dynamic has to be known by the followers. The Cartesian formation allows a stable geometric formation shape. The Frenet formation allows safe formation navigation since the formation shape is adapted according to the leader's trajectory which is safe in static environments. Furthermore, this chapter focused on the analysis to ensure the formation reliability and stability. At this aim, the adaptive constraints, taken by the leader (according to the desired formation shape and its current dynamic) have been proposed to improve the performance of the formation (fast and steady convergence). To deal with the dynamic reconfiguration of the fleet of UGVs, a fully reactive strategies between the UGVs based on suitable smooth switches of the virtual target configurations and inter-target distances were proposed. This strategy avoids the use of predefined trajectories and it can be applied for different situations when the formation has to be modified according to the environment context (dynamic, cluttered, etc.). Furthermore, these strategies takes into account the probable collisions between vehicles as well as the vehicle constraints to ensure safe navigation to reach the new desired formation. Different simulations and experiments using UGVs have shown the reliability, efficiency and flexibility of the proposed strategies for multi-vehicle navigation and dynamic reconfiguration.

GENERAL CONCLUSION AND FUTURE WORKS

GENERAL CONCLUSION

This presented thesis deals with the autonomous navigation of a single and a group of UGVs in formation. The considered navigation task is dedicated to urban transportation system where the UGV has to transport passengers through specific environments. At this aim, the proposed control architectures and methods along this manuscript were developed to guarantee safe, flexible and reliable navigation of a single UGV as well as a group of them with dynamic reconfiguration of the fleet in complex environments (e.g., unknown, cluttered and dynamic areas). Moreover, these proposals can be also extended to different applications of mobile robots from the literature such as exploration, coverage, agriculture, storage, etc.

The proposed control architectures are based on hybrid architecture (central module leading the whole system and elementary reactive behaviors) and Centralized/Decentralized MRS architecture (central entity allows to manage the different tasks of all entities and each entity perform its assigned task using its local sensors). The elements of the proposed control architectures are organized in three layers: **Planning** (selection of waypoints), **Executive** (behavior selection, target assignment and formation set-points) and **Behaviors** (target reaching and obstacle avoidance). The developed methods in each element/block were designed to obtain a generic, reliable and flexible control architecture and online and easy implementation for autonomous navigation of a single or a group of UGVs in unforeseen situations (obstacles) and/or well-know environments (planning).

Several simulations and experiments allow to validate the online application of the proposed control architectures for the navigation of a single and a group of VIPALABs. The main contributions of the thesis can be categorized according the following items:

1. Reliable control architectures for single and multi-vehicle systems.

2. Stable control law for target reaching.

3. Flexible navigation through successive waypoints.

4. Efficient planning for optimal selection of waypoints.

5. Suitable coordination of multi-vehicle navigation in formation.

1/ RELIABLE CONTROL ARCHITECTURES FOR SINGLE AND MULTI-VEHICLE SYSTEMS

The proposed multi-controller architectures have been designed to obtain a distributed and reactive control of each UGV. They break the complexity of the overall tasks (single and multi UGVs' navigations) into a group of reliable and feasible behaviors/controllers. The elementary controllers are *Obstacle avoidance* and *Target reaching*. The *obstacle avoidance* is based on elliptical limit-cycle approach since this approach allows a reactive behavior while using only local information (cf. Section 3.2). This online method uses the local information from range sensors (LIDAR) (Perception block (cf. Section 3.4)). Therefore, online, accurate and stable ellipses which surround the obstacle are obtained to apply the limit-cycle approach for obstacle avoidance. The *Target reaching* allows to guarantee a safe UGV's navigation towards one or more successive (static or dynamic) goals (cf. Section 4.2). The autonomous navigation in cluttered and structured environments is obtained combining these last behaviors using a suitable multi-controller coordination (selection of one controller according to presence of obstacles (cf. Section 3.3.5)).

2/ STABLE CONTROL LAW FOR TARGET REACHING

A single control law for both behaviors (*Obstacle avoidance* and *Target reaching*) in the control architecture has been proposed. These last behaviors have a homogeneous error definition $(e_x, e_y, e_\theta, v_T)$ to be zeroed by the control law (cf. Section 3.3). This control law, in addition to position and orientation, introduces a new parameter related to the orientation of the line that connects the UGV and its target. This new parameter allows to perform static and dynamic target reaching. Therefore, the proposed control law has a satisfactory performance w.r.t. the other control laws (based on path following and trajectory tracking). The stability of this control law is proved using Lyapunov analysis. The feasible target reaching is guaranteed using the relation between the controller parameters $\mathbf{K}$ and the attainable maximal errors $(E_{dis},\ E_\angle)$ (cf. Section 3.5.2).

3/ FLEXIBLE NAVIGATION THROUGH SUCCESSIVE WAYPOINTS

A novel strategy to drive the vehicles through successive waypoints suitably placed in the environment have been presented. This strategy was inspired by human driver behavior where the driver reactively guides his vehicle through specific roads, round-about, etc. without the use of any predefined trajectory. This strategy is combined with the proposed control law which guarantees the smooth and feasible UGV's trajectories using notably successive waypoints. The strategy is flexible in the sense that the UGV can perform different behaviors between to waypoints such as obstacle avoidance.

4/ EFFICIENT PLANNING FOR OPTIMAL SELECTION OF WAYPOINTS

To perform the navigation based on successive waypoints (cf. Section 4.2) three methods for optimal selection of waypoints (number, poses, velocities, etc.) have been proposed. The first method is applied when a specific reference trajectory have been predefined. The proposed method selects the suitable set of waypoints analyzing the orientation change at each point of the trajectory (cf. Subsection 4.3.2). The second and third

methods are applied in a well-known environment (cf. Subsection 4.3.3). They are optimal multi-criteria waypoint selection (OMWS) based on Grid Map (GM) and Expanding Tree (ET). The multi-criteria optimization allows to take into account the safety, feasibility, velocity, steering angle and kinematic constraints of the UGV. OMWS-ET presents more advantages than OMWS-GM and others methods from literature such as less number of waypoints, smoother, safer and shorter trajectories (cf. Subsection 4.3.4).

5/ SUITABLE COORDINATION OF MULTI-VEHICLE NAVIGATION IN FORMATION

The proposed control architecture have been extended to cope with the navigation in formation of a group of UGVs where each UGV has to keep a desired distance and orientation w.r.t. the others with the property of dynamic reconfiguration of the formation according to environmental context (narrow tight corridor or a very cluttered environment). The proposed strategy based on Leader-follower consists in the definition of virtual targets and their configurations are linked to the leader's configuration (cf. Section 5.2). This strategy allows a distributed control since each follower tracks its assigned target while respecting the vehicle's constraints (cf. Section 5.3). The virtual targets can be defined according to current leader's pose (Cartesian frame (rigid geometric shape)) or leader's trajectory (Frenet frame (parallel trajectories to the leader)). Furthermore, the proposed strategies for formation reconfiguration based on smooth switches and movements of the virtual targets allow to each follower to converge towards its dynamic virtual target while maintaining safe inter-vehicle distances (cf. Section 5.4). The stability of the reconfiguration process has been proved using Lyapunov analysis.

FUTURE WORKS

This thesis allows to explore and to extend several research fields in autonomous navigation of a single UGV and a group of them. The most important future works are described below.

1/ RELIABLE CONTROL ARCHITECTURES FOR SINGLE AND MULTI-VEHICLE SYSTEMS

An important study will be the stability of the whole control architecture during the switching phase (between the Target-reaching and obstacle avoidance behaviors and between successive waypoints or obstacles). Moreover, the robustness of the MRS system will be improved while considering leadership change and fault detections. The *Perception* block for online obstacle detection will be improved using camera system or 3D LIDAR. These sensors can allow to discriminate the different obstacles (3D model) or even gives a priority in the selection and the way to avoid obstacles. Furthermore, the proposed control architectures will integrate other methods/blocks based on artificial intelligence (neural networks, fuzzy logic or Markovian process) to enhance the decision process.

2/ STABLE CONTROL LAW FOR TARGET REACHING

The control law block will be extended to considers uncertainties and dynamic modeling of the UGVs. It will allow to obtain a robust control for a safe navigation. Moreover,

a dynamic adaptation of the controller parameters according to adequate performance criteria will be studied to improve the performance of the control law according to the assigned task.

3/ Flexible navigation through successive waypoints

The *Sequential target* block for navigation through successive waypoints will take into account the different traffic signals (lights, pedestrian cross, velocity limits, etc.) for navigation in urban environments. Each waypoint will integrate semantic properties to allow the vehicle to stop or to navigate by adapting its velocity.

4/ Efficient planning for optimal selection of waypoints

The proposed algorithm for optimal selection of the set of waypoints OMWS-ET (Waypoint configuration selection) will be expanded to 3-dimensional space for UAV applications. Moreover, genetic algorithm will be studied to improve the performance of OMWS-ET, mainly according to the most appropriate **K** parameters (cf. Section 3.5).

5/ Suitable coordination of multi-vehicle system

The methods of *Formation parameters* block will unify Cartesian and Frenet formation to allow an automatic selection of the suitable formation according to the application or to the environment. The strategy for formation reconfiguration will consider an optimal allocation of targets to contribute to the fast convergence towards the new desired formation shape.

6/ Experimental works

The localization system will be robust using different sensors based on camera, LIDAR, odometry, etc. to improve vehicle's localization. Moreover, a security system for inaccurate localization will be developed. The methods combining different sensors (camera and LIDAR) to improve the obstacle detection will be implemented. A robust system to deal with delay or fault in communication will be integrated to the whole control architecture. The proposed control architecture with the theoretical perspectives will be analyzed and tested for higher UGV's velocities.

III

ANNEXES

A

MATHEMATICAL FORMULATION OF AN ELLIPSE

This appendix describes the definitions, representations and formulations of an ellipse. The terminology used in this appendix is based mainly on [25, 82].

A.1/ STANDARD REPRESENTATION

An ellipse is defined as a set of points in $\mathbb{R}^2$ plane whose sum of the distances from two fixed point (Focis F_1 and F_2 ($F_1 \neq F_2$)) is a constant $2a \in \mathbb{R}^+$ (cf. Fig. A.1). The set of points (x, y) satisfy the following expression:

$$||(x - x_1, y - y_1)|| + ||(x - x_2, y - y_2)|| = 2a \tag{A.1}$$

where $F_1 = (x_1, y_1)$ and $F_2 = (x_2, y_2)$.

In Fig. A.1, the ellipse has a center at (h, k) and tilt equal to Ω. Some algebraic manipulations of eq. (A.1) leads to the standard form of an ellipse:

$$\left(\frac{x'}{a}\right)^2 + \left(\frac{y'}{b}\right)^2 = 1 \tag{A.2}$$

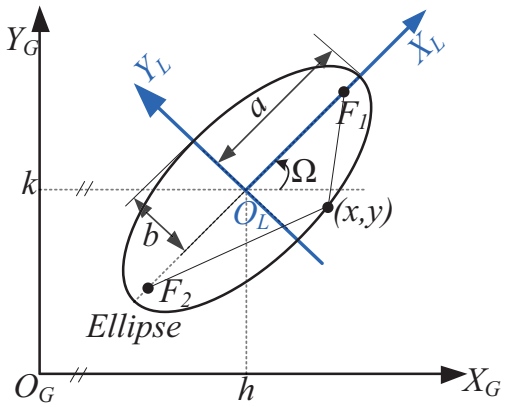


Figure A.1: Ellipse's representation in Global ($X_G Y_G$) and Local ($X_L Y_L$) reference frames.

where (x', y') are the coordinates of the point (x, y) w.r.t. Local $(X_L Y_L)$ reference frame. They are computed as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) \\ -\sin(\Omega) & \cos(\Omega) \end{bmatrix} \begin{bmatrix} x-h \\ y-k \end{bmatrix} \tag{A.3}$$

A.2/ PARAMETRIC REPRESENTATION

The parametric representation $(x(t), y(t))$ can be obtained from standard form (A.2) as follows:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} h \\ k \end{bmatrix} + \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) \\ \sin(\Omega) & \cos(\Omega) \end{bmatrix} \begin{bmatrix} a\cos(t) \\ b\sin(t) \end{bmatrix} \tag{A.4}$$

where $t \in [0, 2\pi]$

A.3/ CONIC REPRESENTATION

The ellipse can be represented by a general quadratic form (substituting eq. (A.3) in eq. (A.2) and expanding all terms). This general equation can describe different conic forms (circle, ellipse, hyperbole and parabola). It is given by:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{A.5}$$

where $A,\ B,\ C,\ D,\ E,\ F \in \mathbb{R}$ An ellipse is define if $B^2 - 4AC < 0$. To avoid degenerated ellipse (imaginary or parallel lines), it is also required:

$$\frac{D^2}{4A} + \frac{E^2}{4C} - F > 0 \tag{A.6}$$

A.4/ CONVERSION BETWEEN REPRESENTATIONS

This section presents the methods to obtain the equivalence between the described representations.

A.4.1/ PARAMETRIC TO CONIC REPRESENTATION

The parameters of eq. (A.4) are used to compute the terms of eq. (A.5) as follows:

$$A = (a\sin(\Omega))^2 + (b\cos(\Omega))^2 \tag{A.7}$$

$$B = -2\cos(\Omega)\sin(\Omega)\left(a^2 - b^2\right) \tag{A.8}$$

$$C = (a\cos(\Omega))^2 + (b\sin(\Omega))^2 \tag{A.9}$$

$$D = -2Ah - kB \tag{A.10}$$

$$E = -2Ck - hB \tag{A.11}$$

$$F = -(ab)^2 + Ah^2 + Bhk + Ck^2 \tag{A.12}$$

A.4.2/ CONIC TO PARAMETRIC REPRESENTATION

The terms of eq. (A.5) are used to obtain the parameters of eq. (A.4) as follows:

$$a = \sqrt{-\frac{det(M_0)}{det(M)\lambda_1}} \tag{A.13}$$

$$b = \sqrt{-\frac{det(M_0)}{det(M)\lambda_2}} \tag{A.14}$$

$$h = \frac{BE - 2CD}{4AC - B^2} \tag{A.15}$$

$$k = \frac{BD - 2AE}{4AC - B^2} \tag{A.16}$$

$$\Omega = \frac{1}{2}\tan\left(\frac{B}{A - C}\right) \tag{A.17}$$

where the matrix M_0 and M are defined as:

$$M_0 = \begin{bmatrix} F & D/2 & E/2 \\ D/2 & A & B/2 \\ E/2 & B/2 & C \end{bmatrix} \tag{A.18}$$

$$M = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix} \tag{A.19}$$

λ_1 and λ_2 are the eigenvalues of M ordered as $|\lambda_1 - A| \leq |\lambda_1 - C|$ (and $|\lambda_2 - A| \leq |\lambda_2 - C|$).

A.5/ COMMON METHODS FOR ENCLOSING DATA WITH AN ELLIPSE

This section shows two common methods in the literature to enclose data with an ellipse.

A.5.1/ FITTING USING LEAST SQUARE METHOD

This approach uses the general conic equation which is given by:

$$f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{A.20}$$

According to the real constants A, B, C, D, E anf F, the analytic equation of the different kind of conics (parabola, ellipse and hyperbole) is obtained. An ellipse is defined if the conic parameters (A.20) satisfy the following condition $B^2 - 4AC < 0$.

The problem is to fit a conic section (A.20) with a set of n points $\{p_i\} = \{(x_i, y_i)\} \mid i = 1, \ldots, n$. As the data are noisy, it is unlikely to find a set of parameters (A, B, C, D, E, F) (except for the trivial solution $A = B = C = D = E = F = 0$) such that $f(x_i, y_i) = 0$. Instead, we will try to estimate them by minimizing some objective function J.

A common practice is to minimize the algebraic distance $f(x_i, y_i)$, the least-square fitting based on algebraic distance [97] is used to minimize the following function:

$$J = \sum_{i=1}^{n} f^2(x_i, y_i) \tag{A.21}$$

Note that there is no justification for using algebraic distance apart from easy implementation. To avoid the trivial solution, $f(x_i, y_i)$ should be normalized. the normalization $A + C = 1$ is used because it is one of the most popular methods proposed in the literature [97], and its implementation is easy.

Using the normalization in (A.20), the set of equations $f(x_i, y_i)$ can be written as:

$$f_i \equiv \mathbf{a}_i\mathbf{s} - \mathbf{b}_i = 0 \tag{A.22}$$

where $\mathbf{s} = [A, B, D, E, F]^T$, $\mathbf{a}_i = [x_i^2 - y_i^2, x_i y_i, x_i, y_i, 1]$ and $\mathbf{b}_i = -y_i^2$. The solution that minimize the functional J is given by

$$\mathbf{s} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{A.23}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^T$ and $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]^T$. This method is known as the pseudo inverse technique.

A.5.2/ COVARIANCE METHOD

This approach is based on the analysis of the relationship between the n points. The covariance of the data and the Mahalanobis Distance (MD) [119] are used. In the field of multivariate calibration, the MD is used for different purposes, namely: for the detection of outliers, the selection of calibration samples from a large set of measurements and for investigating the representativity (matching) between two data sets. In the original variable space, the MD takes into account the correlation in the data, since it is calculated using the inverse of the covariance matrix of the data set of interest.

The principle of covariance approach is given below: First the covariance matrix $\mathbf{C}_x$ is constructed:

$$\mathbf{C}_x = \frac{1}{(n-1)}(\mathbf{X}_c)^T(\mathbf{X}_c), \tag{A.24}$$

where $\mathbf{X}_c$ is the column-centered data matrix $(\mathbf{X} - \bar{\mathbf{X}})$. $\mathbf{X}$ is the data matrix containing n objects in the rows measured for p variables and $\bar{\mathbf{X}}$ is the data mean.

Now, the covariance matrix $\mathbf{C}_x$ is analyzed using the eigenvalues and eigenvectors to obtain the parameters of the ellipse $f(x', y')$ (cf. eq. (3.1)). Furthermore, the Mahalanobis distance is used to select the dimension of the semi-axes (percentage of the data inside the ellipse) [119]. The eigenvalues λ_1, λ_2 $(\lambda_1 > \lambda_2)$ are related to the semi-axes a and b as follows:

$$\begin{aligned} a &= MD_{max}\sqrt{\lambda_1} \\ b &= MD_{max}\sqrt{\lambda_2} \end{aligned} \tag{A.25}$$

where MD_{max} is the maximum Mahalanobis distance to ensure that all the set of data is enclosed by the ellipse. The eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$ are related with the orientation Ω.

$$\Omega = \arctan(\mathbf{v}_2/\mathbf{v}_1) \tag{A.26}$$

The center of the ellipse is the mean of the data $\bar{\mathbf{X}}$ and the maximum semi-axes a is the direction where there are the most important amounts of data.

B

OBSTACLE AVOIDANCE TOOLS

B.1/ DESCRIPTION OF THE OBSTACLE AVOIDANCE ALGORITHM

In what follows, the overall methodology to achieve the used obstacle avoidance algorithm [293] will be described. The algorithm is developed according to stimuli-response principle. To implement this kind of behavior it is important to have to:

- Detection of the hinder obstacle (cf. Section 3.1),
- Assignation of the direction of the avoidance (clockwise or counter-clockwise),
- Definition of an escape criterion to specify when the obstacle is completely avoided.

All these different steps must be followed and applied while guaranteeing that: the robot's trajectory is safe, smooth and it avoids undesirable situations as deadlocks or local minima; and that the stability of the applied control law is guaranteed (cf. Subsection 3.5). The obstacle observation is obtained in real time, the robot does not have any global information about the hinder obstacle nor the ellipse that encloses it. The robot discovers thus at each sample time the shape of the obstacle and increases progressively the knowledge of the enclosing ellipse to obtain a smooth elliptical trajectory (cf. section 3.4). The global information of the obstacle is not related to the used reactive navigation. The necessary steps to carry out the obstacle avoidance algorithm (5) are given below:

1. For each sample time, obtain the distance D_{RO_i} for each potentially disturbing obstacle "i" (cf. Fig. 3.1).
2. Among the set of disturbing obstacles (which can constrain the robot movement to reach the target), choose the closest to the robot (the smallest D_{RO_i} (cf. Fig. 3.1)). This specific obstacle has the following features: (x_{obst}, y_{obst}) center position, a and b as respectively major axis and minor semi-axes and Ω as orientation w.r.t. the global reference frame.
3. After the selection of the closest hinder obstacle, four specific areas (cf. Fig. B.1) are obtained which give the robot's behavior: clockwise or counter-clockwise obstacle avoidance ; repulsive or attractive phase (cf. Algorithm 5). To distinguish between these four areas, it is necessary to:
    - define a specific reference frame which has the following features (cf. Fig. B.1):

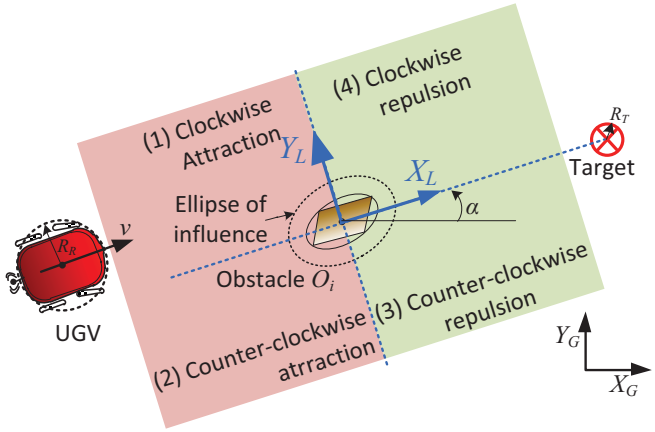


Figure B.1: The four specific areas surrounding the obstacle to avoid [293].

  - the X_O axis connects the center of the obstacle (x_{obst}, y_{obst}) to the center of the target. This axis is oriented towards the target,
  - the Y_O axis is perpendicular to the X_O axis and it is oriented while following trigonometric convention.

- apply the reference frame change of the position robot coordinate $(x, y)_G$ (given in absolute reference frame) towards the reference frame linked to the obstacle $(x, y)_O$. The transformation is achieved while using the following homogeneous transformation:

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_O = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & x_{obst} \\ \sin\alpha & \cos\alpha & 0 & y_{obst} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_G \tag{B.1}$$

Once all necessary perceptions are obtained, one can apply the reactive obstacle avoidance strategy given by Algorithm 5. To obtain the robot set-points, it is necessary to obtain the value of a_{lc} and b_{lc} (cf. Section 3.1) of the orbital ellipse and the direction "clockwise or counter-clockwise" of the limit-cycle to follow. The position (x_O, y_O) gives the configuration $(x,\ y)$ of the robot according to obstacle reference frame. The definition of this specific reference frame provides an accurate means to the robot to know what it must do. In fact, the sign of x_O gives the kind of behavior which must be taken by the robot (attraction or repulsion).

In repulsive phase, the limit-cycle takes an increase value of a'_{lc} and b'_{lc} values to guarantee the trajectory smoothness. The sign of y_O gives the right direction to avoid the obstacle. In fact, if $y_O \geq 0$ then apply clockwise limit-cycle direction else apply counter-clockwise direction. This choice permits to optimize the length of robot trajectory to avoid obstacles. Nevertheless, this direction is forced to the direction taken just before if the obstacle avoidance controller was already active at $t - \delta T$ (δT is the sample time) and it allows to avoid local minimum and dead-end [261].

In algorithm 5, some conflicting situations which are due to local minima or dead ends have to be managed to improve the performance. These specific local and reactive rules are detailed in [261].

Algorithm 5 Obstacle avoidance algorithm [293].

Require: All the features of the closest constrained obstacle.
Ensure: Features of the limit-cycle trajectory to follow.

I) Obtaining the values of a'_{lc} and b'_{lc} of the limit-cycle to follow
if $x_O \leq 0$ **then**
$$\begin{cases} a'_{lc} = a_{lc} - \xi \\ b'_{lc} = b_{lc} - \xi \end{cases}$$ (**Attractive phase**)
▷ with ξ a small constant value as $\xi \ll \delta_E$ which guarantees that the robot do not navigate very closely to the obstacle (cf. Section 3.1).
else
{Escape criterion: go out of the obstacle ellipse of influence with smooth way}
$$\begin{cases} a'_{lc} = a'_{lc} + \xi \\ b'_{lc} = b'_{lc} + \xi \end{cases}$$ (**Repulsive phase**)
end if
II) Obtaining the limit-cycle direction
if obstacle avoidance controller was active at $t - \delta T$ **then**
Apply the same direction already used, equation (3.4) or (3.5) is thus applied.
{This will permit to avoid oscillations and several conflicting situations [261]}
else
The limit-cycle set-point is given by:
$$\begin{aligned} \dot{x} &= m(\bar{C}_{lc}y + 0.5\bar{B}_{lc}x) + \mu x(1 - \bar{A}_{lc}x^2 - \bar{B}_{lc}xy - \bar{C}_{lc}y^2) \\ \dot{y} &= -m(\bar{A}_{lc}x + 0.5\bar{B}_{lc}y) + \mu y(1 - \bar{A}_{lc}x^2 - \bar{B}_{lc}xy - \bar{C}_{lc}y^2) \end{aligned}$$
end if

B.2/ RAPIDLY EXPLORING RANDOM TREE (RRT)

This section briefly summarize the RRT* algorithm based on [301, 426]. RRT* is an incremental sampling based algorithm which finds an initial path very quickly. The path is later optimized as the execution takes place [301].

The objective of RRT* is to find out an input u: $[0 : T] \in U$ that yields a feasible path $x(t) \in$ C-space$_{free}$ that starts from initial position $x(0)$ to $x(T)$ =goal while following the UGV's constraints. While finding this solution, RRT* maintains a tree $T = (V, E)$ of vertices V sampled from the C-space$_{free}$ and edges E that connect these vertices together. The pseudocode of RRT* is explained in Algorithm 6.

At first in the Algorithm 6, a sample x_{rand} is placed randomly in the C-space$_{free}$. Then, the nearest node $x_{nearest}$ to x_{rand} is checked for in the entire configuration space. A node x_{new} is placed at a distance Δx from the nearest node $x_{nearest}$ in the line of direction of x_{rand}. Then, the trajectory path x_{new} is checked if it is free of obstacles. If the trajectory is obstacle free then a ball of radius $\beta(\log(n)/n)$ around x_{new} is checked for near nodes x_{near}. Among this set of nodes, the node that gives the least cost from the starting point to x_{new} is selected as the parent of x_{new}. Once the parent is selected, rewiring takes place. The costs of all the nodes inside this ball around x_{new} is calculated through x_{new}. If this cost is less than the previous cost for any node then that particular node is disconnected from its old parent and is connected to x_{new} as its parent [301].

RRT* is a landmark sampling based algorithm to approach an optimal solution ensuring asymptotic optimality, apart from probabilistic completeness, as opposed to its predeces-

Algorithm 6 Pseudocode of RRT*

Require: Initial $x(0)$ and final $x(T)$ positions
Ensure: Tree $T = (V, E)$ from $x(0)$ to $x(T)$
Initialize Tree $T = (V, E)$
Insert node $x(0)$ to Tree $T = (V, E)$
for $i = 0$ to N_{iter} **do**
    Choose x_{rand} from C-space$_{free}$
    Choose $x_{nearest}$ from $T = (V, E)$
    Compute u_{new} which drives the UGV from $x_{nearest}$ to x_{new} ▷ x_{new} is the closest state to x_{rand}
    if $x_{new} \in$ C-space$_{free}$ **then**
        Obtain neighbor x_{near} of x_{new}
        Select the closest node x_{min} (parent) among $x_{nearest}$ and x_{new} to x_{near}
        Insert vertice x_{new} and the edge $x_{new} - x_{min}$ in $T = (V, E)$
        Rewiring $T = (V, E)$ if the cost through x_{new} is less than older costs.
    end if
end for
Obtain the $T = (V, E)$ from $x(0)$ to $x(T)$

sor RRT (and its various other improved versions [292]).

B.3/ COLLISION BETWEEN UGVS AND OBSTACLES

Collisions between UGVs can occur during the reconfiguration phase of the group of UGVs. To address this collision risk, we use a penalty function acting on the linear velocity of the UGVs [294], [5]. Moreover, this reduced velocity of UGVs allows to obtain a smooth and less oscillating vehicles' movements. Each UGV is enclosed by two circle C_{int} and C_{ext} with respectively radius R_{int} and R_{ext} ($R_{int} < R_{ext}$) (cf. Fig. B.2). The collision occurs when the distance d_{ij} between UGV_i and UGV_j are less than R_{int}. Hence, the penalty function ψ_i^j for the UGV_i w.r.t. the UGV_j is defined as:

$$\psi_i^j = \begin{cases} 1 & \text{if } d_{ij} \geq R_{ext} \\ (d_{ij} - R_{int_i})/(R_{ext} - R_{int_i}) & \text{if } R_{int_i} < d_{ij} < R_{ext} \\ 0 & \text{if } d_{ij} \leq R_{int_i} \end{cases} \tag{B.2}$$

The modified linear velocity of the UGV_i is then given by:

$$\bar{v}_j = v_j \psi_i^j \tag{B.3}$$

Using the definition of R_{int_i} (where $R_{int_i} \neq R_{int_j}$), it is guaranteed that two UGVs do not stop simultaneously. Indeed, if the UGVs have the same R_{int_i} we can observe local minima in certain configurations, in fact, when $d_{ij} < R_{int_i}$ then $\psi_i^j = \psi_j^i = 0$ and the robots are stopped at the same time. R_{ext} is designed according to communication constraints (latency) and localization errors (GPS). This penalty function can be straightforward integrated to our control architecture (cf. Fig. 5.2) by adding a block after the output of the *Control law* block (cf. Fig. B.3).

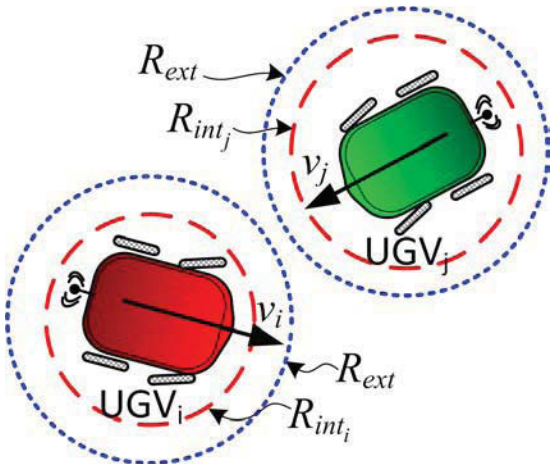


Figure B.2: Representation of circles C_{int_i} and C_{ext} of each UGV.

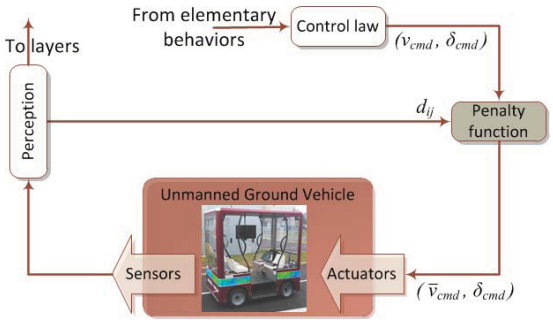


Figure B.3: Integration of the penalty function in the proposed architecture.

C

METHODS OF THE PERCEPTION BLOCK

This appendix details the used functions and methodologies of *Filtering* and *Segmentation* blocks in the *Perception* block of proposed control architecture (cf. Section 3.4).

C.1/ ITERATIVE CLOSEST POINT

This section describes the Iterative Closest Point (ICP) algorithm. ICP is a powerful algorithm for calculating the displacement between scans [246]. The objective of this algorithm is to find the translation **T** and rotation **R** that minimizes the sum of squared error between two corresponding point set $\mathbf{X} = \{x_1, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, \ldots, y_n\}$ as follows (cf. Fig. C.1):

$$E(\mathbf{R}, \mathbf{T}) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \mathbf{R}y_i - \mathbf{T}\| \tag{C.1}$$

In order to address the issue of point correspondence, the ICP algorithm iteratively performs the following steps:

1. Matching: The nearest neighbor of every data point in the model point set is found.
2. Minimization: The error E is minimized with translational **T** and rotational **R** matrix.
3. Transformation: Data points are transformed using **R** and **T** matrix.

Fig. C.2 shows the flowchart of the ICP algorithm. The initial error value E (cf. eq. C.1) is set to infinity to start the loop. E_{th} is the threshold error value given by the user to indicate the convergence of ICP algorithm. The major problem is to determine the correct data associations (matching). Given the correct data associations, the transformation can be computed efficiently using SVD (Singular Value Decomposition) [295]. The algorithm is terminated based on the number of iterations or the relative change in the error metric.

C.2/ FILTERING OF PERCEPTION SENSORS

This block allows to erase the outliers and noises from the range sensor. The method is based on the Iterative Closest Point (ICP) algorithm (cf. Fig. C.3). The ICP is a simple

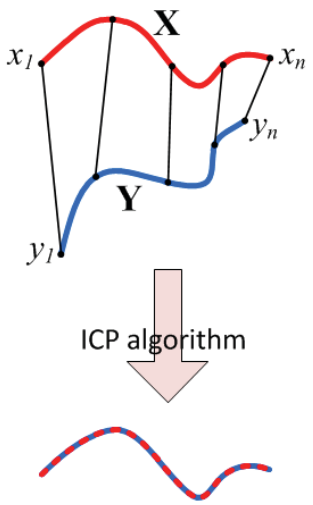


Figure C.1: Example of application of ICP algorithm between two data sets **X** and **Y**.

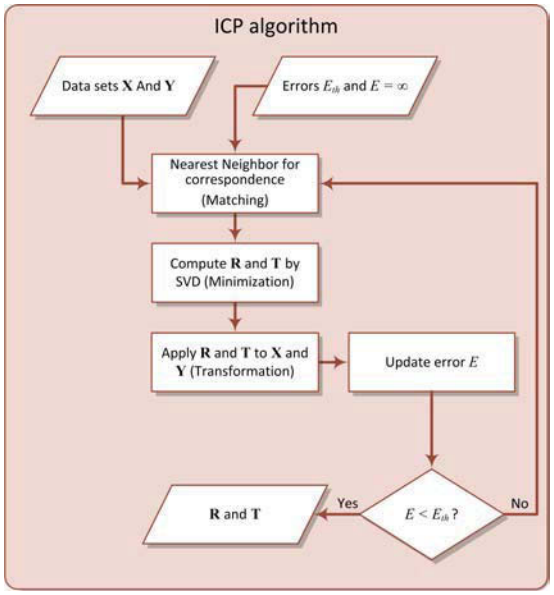


Figure C.2: Flowchart of ICP algorithm.

algorithm used to align two clouds of points with unknown correspondences (cf. Section C.1).

Fig. C.3 shows the flowchart of the Filtering block. The current UGV's pose is used to transform the range data (D_{r_i}, β_i), given in polar coordinates, to a Global reference frame $X_G Y_G$ (cf. Fig. 3.9) using the following equation:

$$\mathbf{r}_i = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} D_{r_i}\cos(\beta_i) + h_r \\ D_{r_i}\sin(\beta_i) + k_r \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{C.2}$$

where $(h_r,\ k_r)$ are the coordinates of the range sensor's center w.r.t. Local reference frame of the UGV $X_m Y_m$ (cf. Fig 3.5). $(x,\ y,\ \theta)$ is the current UGV's pose.

The distance thresholds D_{th} and D_{stop} are used to focus on the meaningful distance information. They are given according to the current UGV's velocity v as follows:

$$D_{th} = D_{max}\left((1 - k_{th})\frac{v}{v_{max}} + k_{th}\right) \tag{C.3}$$

$$D_{stop} = D_{min}\left((k_{stop} - 1)\frac{v}{v_{max}} + 1\right) \tag{C.4}$$

where $k_{th},\ k_{stop} \in \mathbb{R}^+$ with $D_{min}/D_{max} < k_{th} < 1$ and $1 < k_{stop} < D_{max}/D_{min}$. These relations guarantees $D_{th} > D_{stop}$. D_{max} and D_{min} are respectively the maximum detection distance of the sensor and the minimum allowed distance to an obstacle before to collide with it. v_{max} and v are respectively the maximum and current UGV's velocity. β_{th}^- and β_{th}^+ are limit angles for the opening angle to focus on the meaningful information.

The loop of ICP algorithm between current reading **R** and $\mathbf{R}_{tmp}$ allows to eliminate the noise, outliers and possible false data. $\mathbf{R}_{tmp}$ is updated at each comparison for n_{ICP} times. $n_{ICP} \in \mathbb{Z}^+$ indicates the number of reading to be used for the comparison. The ICP algorithm between $\mathbf{R}_{tmp}$ and $\mathbf{R}_{buffer}$ allows to save and update the data from $\mathbf{R}_{tmp}$ in $\mathbf{R}_{buffer}$.

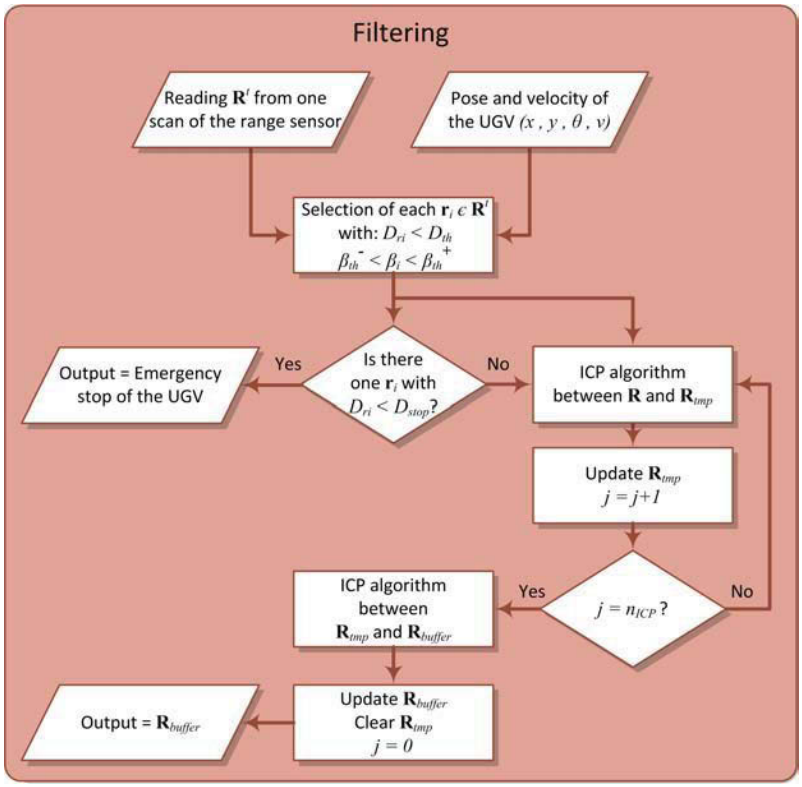
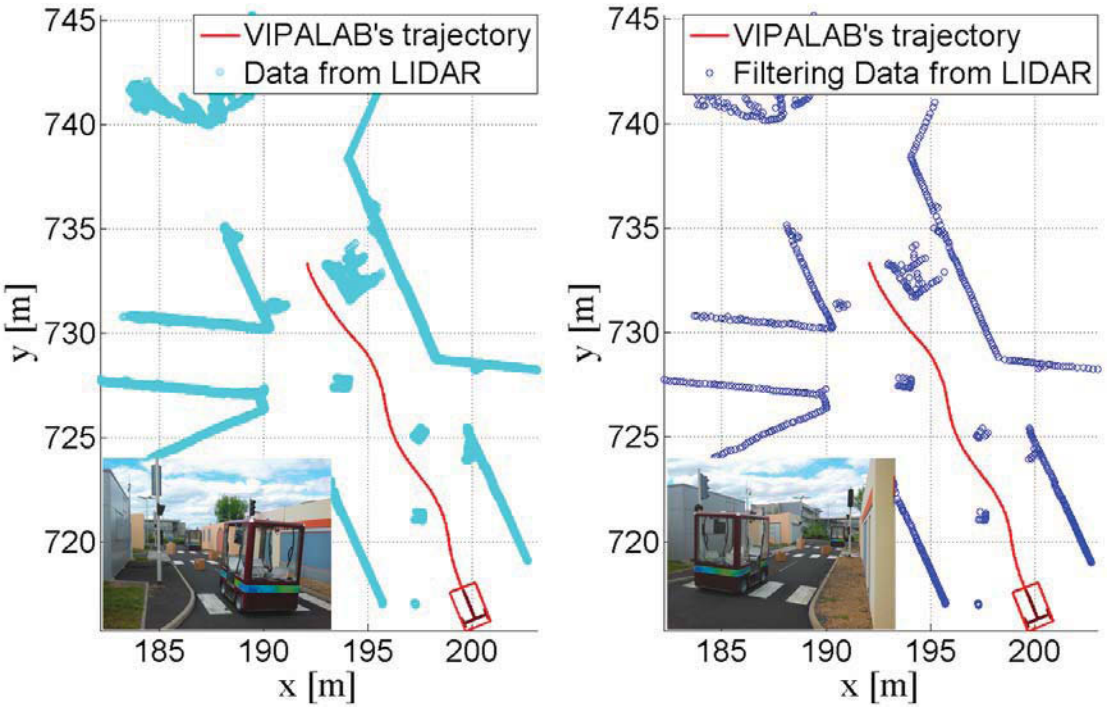


Figure C.3: Flowchart of the Filtering block.



(a) Number of points = 1696

(b) Number of points = 428

Figure C.4: a) LIDAR data without filtering process and b) with filtering process.

Fig. C.4 shows an example of an online application of the Filtering process using a UGV with a LIDAR sensor. The UGV's pose is given by a GPS-RTK and a IMU. It is observed that the Filtering block allows to obtain an accurate information while reducing the problems of inaccuracy and noise.

C.3/ SEGMENTATION METHODS

In computer vision, segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). In this field, the goal of segmentation is to simplify and/or change the representation of an image into something which is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) into images.

In this thesis, segmentation is a fundamental issue in processing of point cloud data acquired by range sensors. The quality of segmentation determines largely the success of information retrieval. The objective of the segmentation of point clouds from range sensors is to spatially group points with similar properties into homogeneous regions or segments (cf. Fig. C.5). For instance, the segmentation is used to detect buildings in [275].

The presented segmentation method is based on [299]. This method consists in grouping points detected by range sensors which correspond to the same obstacle. This phase is usually completed by considering two consecutive points as belonging to the same obstacle if they are not separated by a distance Δd greater than a distance threshold Δd_{th}, otherwise, one points belongs to a new obstacle (cf. Fig. C.5) [137, 206]. This threshold depends on the distance between the laser scanner and the obstacle detected as follows:

$$\Delta d_{th}(r_i, r_{i+1}) = s_0 + s_1 \min\left(D_{r_i}, D_{r_{i+1}}\right) \tag{C.5}$$

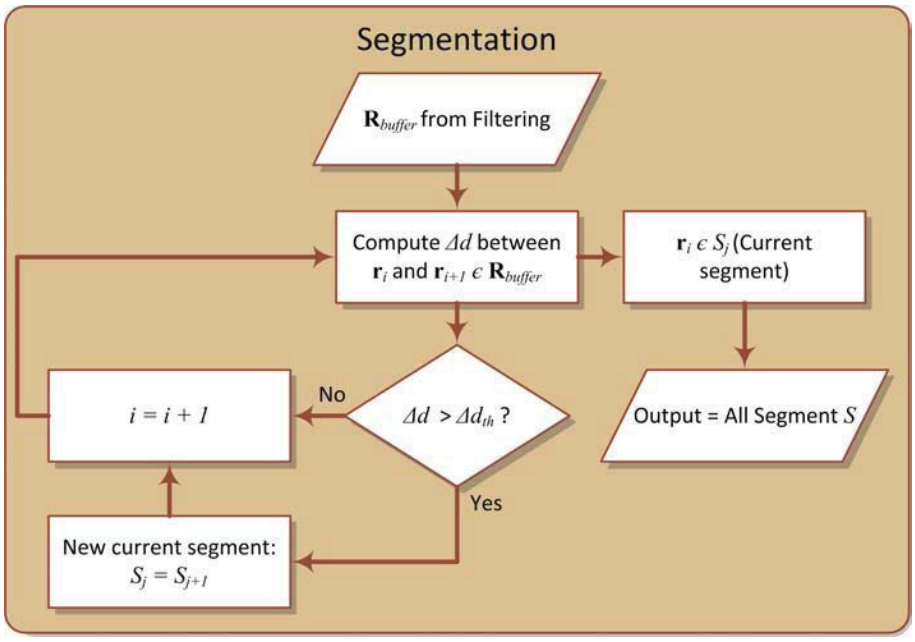


Figure C.5: Flowchart of the Segmentation block.

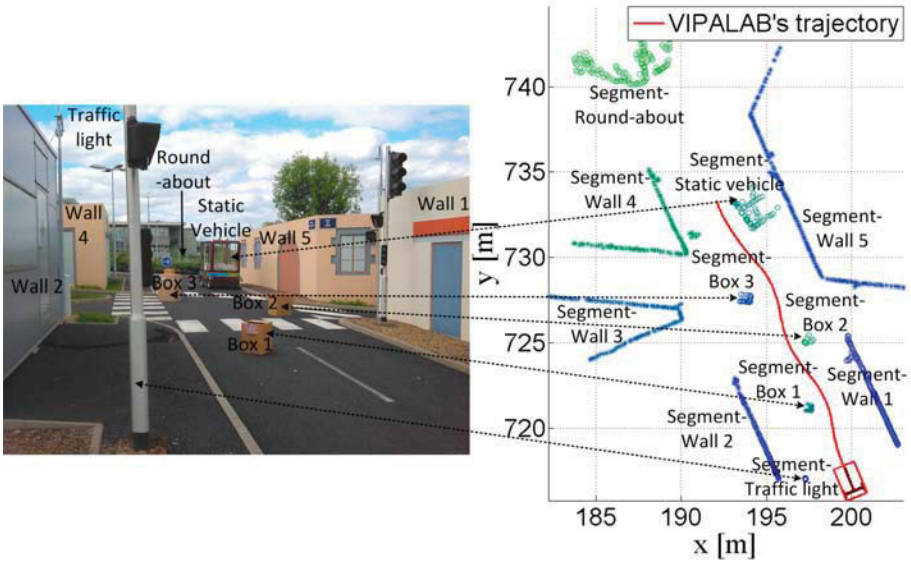


Figure C.6: Example of the application of the segmentation method. Each segment is represented by different colors.

where $s_0 \in \mathbb{R}^+$ is a constant value used for noise reduction, and s_1 can be chosen as:

$$s_1 \geq \sqrt{2 - 2\cos(\Delta\beta)} \tag{C.6}$$

where $\Delta\beta$ is the angular resolution of the range sensor.

The drawback of this method is the accurate categorization of points close to each other but belonging to different objects. Fig. C.6 shows an example of the application of presented segmentation method using data from a LIDAR sensor which has a resolution equal to $\Delta\beta = 1^\circ$.

D

ANALYTICAL SOLUTION OF CONTROL LAW EQUATIONS FOR SAFE TARGET REACHING

This annex describes the solution of simplified differential equation given by eq. (3.41):

$$\dot{e}_\theta = -\frac{K_x(K_d d + K_o K_\theta)}{K_o \cos^2(e_\theta)}\left[K_d d + K_o K_\theta \sin^2(e_\theta)\right]\sin(e_\theta) \tag{D.1}$$

The defined notations c_1, c_2 and c_3 are given by eq. (3.42) as:

$$\begin{cases} c_1 = & K_d d \\ c_2 = & K_o K_\theta \\ c_3 = & \sqrt{(c_1 c_2^{-1} + 1)} \end{cases} \tag{D.2}$$

These notations are replaced in eq. (D.1) to obtain:

$$\frac{de_\theta}{dt} = -\frac{K_x(c_1 + c_2)}{K_o \cos^2(e_\theta)}\left[c_1 + c_2 \sin^2(e_\theta)\right]\sin(e_\theta) \tag{D.3}$$

while reorganizing eq. (D.3), it is obtained:

$$\frac{\cos^2(e_\theta)}{\left[c_1 + c_2 \sin^2(e_\theta)\right]\sin(e_\theta)} de_\theta = -\frac{K_x c_2 (c_1 c_2^{-1} + 1) dt}{K_o} \tag{D.4}$$

Eq. (D.4) can be written as:

$$\frac{\cos^2(e_\theta)}{c_1 \sin(e_\theta)} - \frac{c_2 \sin(e_\theta)\cos^2(e_\theta)}{c_1\left[c_1 + c_2 \sin^2(e_\theta)\right]} de_\theta = -\frac{K_x(c_1 + c_2)dt}{K_o} \tag{D.5}$$

Using trigonometrical properties in eq. D.5, it is obtained:

$$\frac{\cos^2(e_\theta)de_\theta}{c_1 \sin(e_\theta)} - \frac{\sin(e_\theta)\cos^2(e_\theta)de_\theta}{c_1\left[c_1 c_2^{-1} + 1 - \cos^2(e_\theta)\right]} = -\frac{K_x c_2 (c_1 c_2^{-1} + 1)dt}{K_o} \tag{D.6}$$

Regrouping terms in eq. (D.6), it can be written as:

$$\frac{de_\theta}{\sin(e_\theta)} + \frac{c_3 d\cos(e_\theta)}{2\left[c_3 + \cos(e_\theta)\right]} + \frac{c_3 d\cos(e_\theta)}{2\left[c_3 - \cos(e_\theta)\right]} = -\frac{K_x c_1 c_2 c_3^2 dt}{K_o} \tag{D.7}$$

Finally, integrating eq. (D.7), it is obtained eq. (eq:diffequaResolTTx) ≡ (3.43):

$$\int \frac{de_\theta}{\sin(e_\theta)} + \int \frac{c_3 d\cos(e_\theta)}{2\left[c_3 + \cos(e_\theta)\right]} + \int \frac{c_3 d\cos(e_\theta)}{2\left[c_3 - \cos(e_\theta)\right]} = -\int \frac{K_x c_1 c_2 c_3^2 dt}{K_o}$$

$$\ln\left(\tan(e_\theta/2)\right) + c_3/2\ln\left(c_3 + \cos(e_\theta)\right) - c_3/2\ln\left(c_3 - \cos(e_\theta)\right) = -\frac{K_x c_1 c_2 c_3^2}{K_o} t$$

$$\ln\left[\tan\left(\frac{e_\theta}{2}\right)\left(\frac{c_3 + \cos(e_\theta)}{c_3 - \cos(e_\theta)}\right)^{c_3/2}\right] = -\frac{K_x c_1 c_2}{K_o} c_3^2 t \qquad \text{(D.8)}$$

E

CONCEPTS AND NOTIONS OF SYSTEM STABILITY

E.1/ DEFINITION OF STABILITY OF A SYSTEM

This appendix describes the definitions of stability applied to a dynamic system. The system is represented by the following state equation:

$$\dot{x} = f(x) \tag{E.1}$$

where $x \in \mathbb{R}^n$ represents the state of the system. The equilibrium point is assumed at $x = 0$.

- **Simple stability:** Let us consider that the initial time of the system is at $t_0 = 0$. Therefore, the origin point is stable if it satisfies the following expression:

$$\forall \epsilon > 0, \exists\, \delta > 0 : |x(0)| \leq \delta \Rightarrow |x(t)| \leq \epsilon \qquad \forall t \geq 0 \tag{E.2}$$

- **Asymptotic stability:** The system is asymptotically stable if it is stable and δ can be chosen such as:

$$|x(0)| \leq \delta \Rightarrow \lim_{t \to \infty} x(t) \to 0 \tag{E.3}$$

  If eq. (E.3) is satisfied for all δ, then the system is globally asymptotically stable.

- **Exponential stability:** The system is exponentially stable if it satisfies:

$$\exists\, \delta > 0, c > 0, \lambda > 0 : |x(0)| \leq \delta \Rightarrow |x(t)| \leq c\,|x(0)|\, e^{-\lambda t} \qquad \forall t \geq 0 \tag{E.4}$$

E.2/ STABILITY IN THE SENSE OF LYAPUNOV

E.2.1/ FIRST METHOD: INDIRECT METHOD

The first method of Lyapunov is based on the analysis of linearization of the $f(x)$ system around its equilibrium point. This method consists of analyzing the eigenvalues $\lambda_i(J)$ of the Jacobian matrix J at its equilibrium point:

$$J = \frac{\partial f}{\partial x}(0) \tag{E.5}$$

The properties of the stability of systems are expressed as follows:

Theorem 2. ***First method of Lyapunov:***

1. *If all eigenvalues of the J matrix have a strictly negative real part, then the system is exponentially stable.*
2. *If the J matrix has at least one eigenvalue with a strictly positive real part, then the system is unstable.*

If the system has at least one eigenvalue with a zero real part and any eigenvalue with a strictly positive real part, then any conclusion about the stability could be obtained. The system can be analyzed by the second method of Lyapunov.

E.2.2/ SECOND METHOD: DIRECT METHOD

This method consists of an mathematical interpretation of an elementary observation: If the total energy of a system decreases/dissipates along time continuously, then the system tends to an equilibrium point, i.e., the system is stable. The idea is thus to find a temporal positive define function which allows to have always a define negative derivative. This direct method is summarized in the next theorem

Theorem 3. ***Second method of Lyapunov:*** *The equilibrium point is stable if exists a V function continuously differentiable and its derivative denoted by $\dot{V}$ satisfies:*

1. $V(0) = 0,$
2. $V(x) > 0 \quad \forall x \neq 0,$
3. $\dot{V}(x) \leq 0 \quad \forall x \neq 0,$

If condition (3) is replaced by $\dot{V}(x) < 0$ then the system is asymptotically stable.

E.3/ ELLIPTICAL LIMIT-CYCLE

In the dynamical systems, there exists attractor and repellor points, i.e., stable or unstable nodes. Another type of attractor or repellor for nonlinear systems are the limit-cycles.

In what follows, it will be show that the differential equations (3.4) and (3.5) described in Section 3.1 represent a dynamical system which has a stable elliptical limit-cycle. Moreover, it is easily seen that the only equilibrium point of the system occurs at the origin. The following Lyapunov function is considered:

$$V = \left(1 - A_{lc}x_s^2 - B_{lc}x_s y_s - C_{lc}y_s^2\right)^2 \tag{E.6}$$

By considering the rate of change w.r.t. time of eq. (E.6), it is obtained:

$$\begin{aligned}
\dot{V} =& \frac{\partial V}{\partial x}\dot{x} + \frac{\partial V}{\partial y}\dot{y} \\
=& -2(2A_{lc}x_s + B_{lc}y_s)V^{1/2}\left[\ (C_{lc}y_s + \frac{B_{lc}}{2}x_s) + x_s V^{1/2}\right] \\
& -2(2C_{lc}y_s + B_{lc}x_s)V^{1/2}\left[-(A_{lc}x_s + \frac{B_{lc}}{2}y_s) + y_s V^{1/2}\right] \\
=& -2V^{1/2}\left[(2A_{lc}x_s + B_{lc}y_s)(C_{lc}y_s + \frac{B_{lc}}{2}x_s) - (2C_{lc}y_s + B_{lc}x_s)(A_{lc}x_s + \frac{B_{lc}}{2}y_s)\right] \\
& -2V\left[x_s(2A_{lc}x_s + B_{lc}y_s) + y_s(2C_{lc}y_s + B_{lc}x_s)\right] \\
=& -4V\left[A_{lc}x_s^2 + C_{lc}y_s^2 + B_{lc}x_s y_s\right] \\
=& -4\left(1 - A_{lc}x_s^2 - B_{lc}x_s y_s - C_{lc}y_s^2\right)^2\left[A_{lc}x_s^2 + B_{lc}x_s y_s + C_{lc}y_s^2\right]
\end{aligned}$$

Thus, it can be noted that V decreases on the trajectories given by eq. (3.4) and (3.5) except at the origin where it clearly has a relative maximum and on the ellipse $A_{lc}x_s^2 - B_{lc}x_s y_s - C_{lc}y_s^2 = 1$ where it is identically zero. The term $A_{lc}x_s^2 + B_{lc}x_s y_s + C_{lc}y_s^2$ is always positive since it can be represented in function of only quadratic terms (cf. standard form of an ellipse (cf. eq. A.2) in Appendix A)

Outside the ellipse the function is decreasing towards the ellipse. Inside the ellipse the function is decreasing towards the ellipse again. Thus the trajectories of the system must move towards the ellipse from points outside or inside of it for increasing t.

It can be constructed a trapping region by taking any contour inside the ellipse as the inner boundary of an annular region together with any contour outside the ellipse as the outer boundary. Hence by the Poincaré-Bendixson theorem [147] since there are no equilibrium points within the annular region there must be a stable limit cycle within the region. Note also that the contours can be taken as close as possible to the ellipse, therefore, the limit cycle must be the ellipse.

F

PROCRUSTES DISTANCES

The Procrustes shape distance [31] is used as criteria to quantify the distortion between the desired formation **F** and the real formation shape [345]. Basically, the Procrustes distance P_d is a least-square type shape metric that require aligned shapes with one-to-one point correspondence. The alignment part involves four steps:

1. Compute the centroid of each shape.
2. Re-scale each shape to have equal size.
3. Align w.r.t. position the two shapes at their centroids.
4. Align w.r.t. orientation by rotation.

The P_d is given by square root of the summed squared difference between the positions of the vertex in two optimally superimposed configurations at centroid size (cf. Fig. F.1). It is given by:

$$P_d^2 = \sum_{i=1}^{n} \left[(x_{i_1} - x_{i_2})^2 + (y_{i_1} - y_{i_2})^2 \right] \tag{F.1}$$

where (x_{i_1}, y_{i_1}) and (x_{i_2}, y_{i_2}), are respectively the aligned coordinates of the vertex i of the first and second shapes. The maximum distance between vertex Dn_{max} is defined as:

$$Dn_{max}^2 = \max\{(x_{i_1} - x_{i_2})^2 + (y_{i_1} - y_{i_2})^2, i = 1, \ldots, n\} \tag{F.2}$$

Additionally, the $\mathcal{L}_2$ norm can be applied to the evolution of P_d and Dn_{max}. The norm is given by:

$$\mathcal{L}_2(x) = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} \|x\|_2^2 dt} \tag{F.3}$$

where t_0 and t_f are respectively the initial and final time of the variable x.

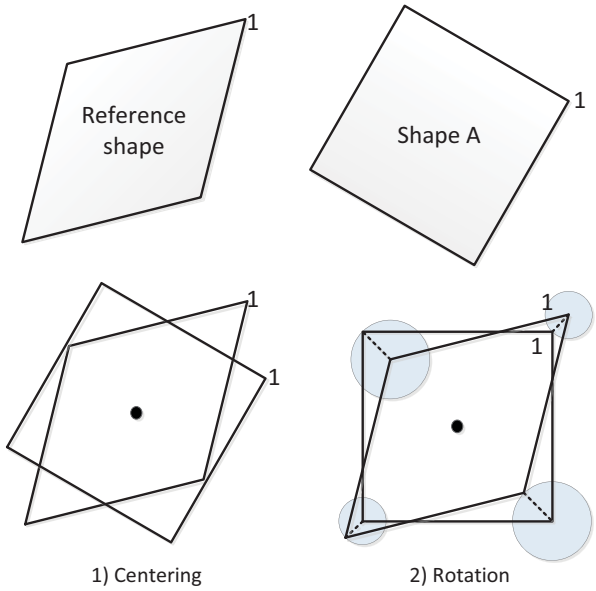


Figure F.1: Schema for compute the Procrustes distances.

G

SIMULATION AND EXPERIMENTAL TESTBED

This appendix described the main elements for the implementation of the proposed control architecture for navigation in formation of a group of UGVs. The elements of the used platform and the structured environment are described in the next section. Section G.2 presents the used simulator software and the interface to perceive and command the UGVs. The structure of the implementation of the control architecture embedded in each UGV is explained in Section G.3.

G.1/ PLATFORM AND SCENARIOS

The Véhicule Individuel Public et Autonome (VIPALAB) from APOJEE company [357] is a platform dedicated to the development of autonomous vehicles. This urban electric vehicle VIPALAB is used to implement several proposed control architectures for autonomous navigation of one and a group of UGVs [4, 8, 9]. Some specifications of VIPALAB are shown in Table G.1 [357]. This vehicle carries different embedded proprioceptive and exteroceptive sensors such as cameras, odometers, IMUs, steering angle sensor, an RTK-GPS, a Wi-Fi communication system and a computer (more details are given in [357]). The VIPALAB can be controlled using the on-board computer (through CAN protocol) or the wired control panel attached to the vehicle.

Table G.1: VIPALAB platform

VIPALAB	Elements	Description
	Chassis	(l, w, h)= $(1.96, 1.30, 2.11)\ m$
	Weight	$400\ kg$ (without batteries)
	Motor	Triphase $3x28\ V$, $4\ KW$
	Break	Integrated to the motor
	Maximum speed	$20\ km/h\ (\approx 5.5\ m/s)$
	Batteries	8 batteries $12\ V$, $80\ Ah$
	Autonomy	3 hours at full charge
	Computer	Intel Core i7, CPU:$1.73\ GHz$ RAM:8Go OS(32bits):Ubuntu12.04

G.1.1/ USEFUL SENSORS

Several sensors are mounted in the VIPALAB to obtain vehicle's interoceptive and exteroceptive information about the UGV or its environment [357] (cf. Fig. G.1). The main sensors used for experiments of the proposed control architectures are described below:

- Rear wheels odometry: Each rear wheel is equipped with an encoder in the form of a ring gear providing 64 *top* per wheel revolution. The data provided by wheel encoders can be used to obtain displacement up to an accuracy of $2\ cm$ and hence the velocity can be estimated.
- Steering angle: The steering angle is obtained by reading the analog value of a potentiometer. The saturated range of $\pm 30°$ of the steering angle are digitized using a 12-bit ADC, which gives a theoretical resolution of 0.015 degrees.
- Motor odometry: This proprioceptive sensor provides data from the motor encoder related to translational velocity with an accuracy of $0.1\ m/s$. Translational velocity and steering angle are combined using an Ackerman model in dead reckoning ($40\ cm$ in diameter of the wheels) to obtain the VIPALAB's trajectory. All proprioceptive measurements (odometry and steering angle) are accessible through a CAN gateway at a framerate of $50Hz$.
- Inertial Measurement Unit (IMU): Xsens Mti10 IMU sensor is embedded at the middle of the rear-axes of the VIPALAB. The IMU combines the information of accelerometers and gyroscopes to provide the angular and linear acceleration information along the three axes. It allows to obtain the yaw information (rotation along the axis perpendicular to the ground plane). The output frequency is up to $2\ Khz$ with an accuracy of $0.2°/s$ and latency is less than $2\ ms$.

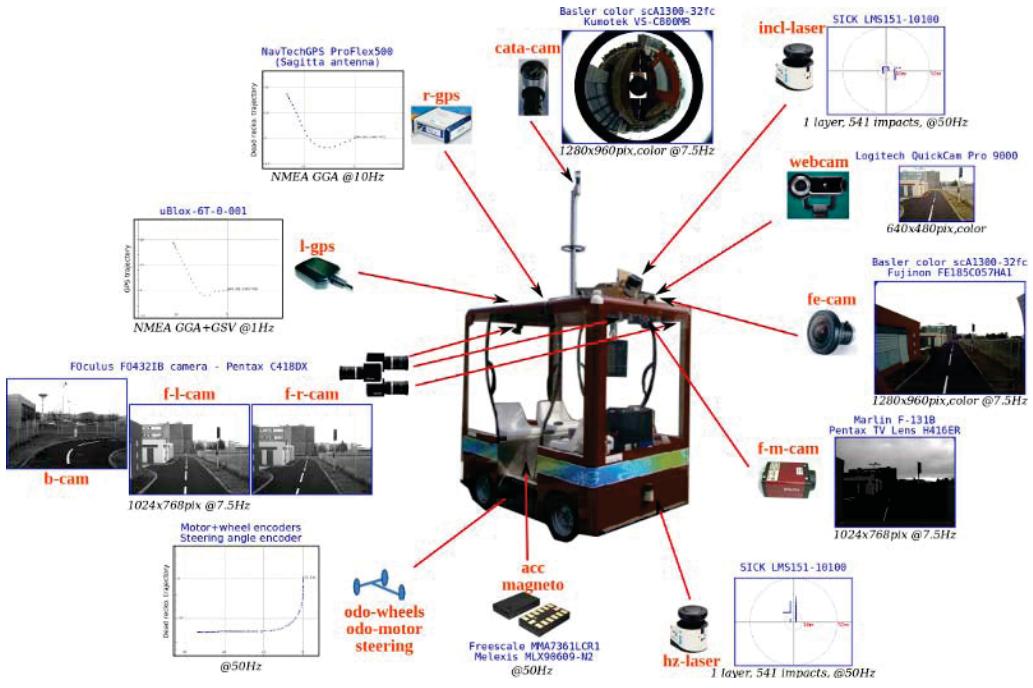


Figure G.1: VIPALAB with all sensors with their mounting locations and characteristics.

- RTK-GPS: ProFlex500 RTK GPS receiver (antenna) is mounted on the top of the VIPALAB. The data acquired with GPS receiver coupled with a Sagitta Magellan GPS base station are considered as the ground truth. This localization system provides an absolute localization measurement with an accuracy within $2\ cm$ at a frequency of $10\ Hz$.

- Horizontal range sensor: SICK LMS151-10100 planar range sensor is mounted horizontally (parallel to the ground plane) on the frontal facade of the VIPALAB. This LIDAR allows to scan a horizontal plane at $40\ cm$ from the ground plane. The information (output) given by the LIDAR are distance and angle (polar coordinates (r, θ)). The characteristic of the LIDAR are range in $[0,\ 50]\ m$, angle in $[-45°, 225°]$, resolution $0.5°$ and frequency of $50\ Hz$.

- Wi-Fi communication: The wireless system uses a Laguna base system (2xITS $5.9\ GHz$ $802.11p$ radios). It is universal for vehicle and infrastructure functionality.

G.1.2/ STRUCTURED ENVIRONMENT

The test platform named Plate-forme d'Auvergne pour Véhicules INtelligents (PAVIN) is located at Campus Cézeaux of Blaise Pascal University in Clermont-Ferrand (cf. Fig. G.2). PAVIN lies between the faculties of ISIMA and Polytech. PAVIN is an artificial environment composed by two areas (urban and rural areas) which have a total ground surface of $5.000\ m^2$ which serves as a testbed for mobile robotic applications. The urban area has a trajectory of $317\ m$ containing scaled street with several traffic junctions and roundabouts with traffic sign boards and lights wherever necessary. Moreover, building facades on both sides, vegetation and street furniture are set to bring to a whole scene. The rural area has a trajectory of $264\ m$ with unpaved roads, grass and mud on the roadsides. In addition, the whole area is covered by a wireless network and a DGPS base station [357].

Although PAVIN is a small scale environment, it stands as an ideal platform for evaluating algorithms related to autonomous driving such as navigation, road detection, traffic signal detection, etc. A 2D and a textured 3D model of PAVIN environment geo-referenced with high-precision GPS data are available in [357].

Figure G.2: PAVIN experimental platform (Clermont-Ferrand, France).

G.2/ SOFTWARE TOOLS

This section presents two used software for the real implementation of the proposed control architecture for the navigation of a single and a group of VIPALABs. The first software is COBAYE simulator which allows to test algorithms (programming code) in virtual model of VIPALAB with its sensors. The second software is Effibox middleware which allows to manage the information from real sensors and also it is compatible with the programming code developed in COBAYE simulator.

G.2.1/ COBAYE SIMULATOR

The COBAYE® simulator is a multi-variable platform which allows to recreate several scenarios with different types of robots. This simulator was developed by 4D-Virtualiz [413]. COBAYE editor allows configuring all necessary parameters to create the desired scene. Several elements can be added such as pedestrians, drums, trailers, boxes, etc. Even, pedestrians can be animated with a random movement to simulate their behaviors in public areas. Furthermore, COBAYE editor allows to simulate several sensors for navigation such as cameras, GPS, IMU, odometers, etc. Once sensors are added to the scenario, COBAYE editor allows to configure their desired position in the vehicle and technical specifications.

G.2.2/ EFFIBOX MIDDLEWARE

Effibox® is a middleware which access and record process from real time sensor data. This middleware was developed by Effidence [416]. The main characteristic are multi-thread, open C++ API (application programming interface), several certified commercial sensors (camera, GPS, LIDAR, Sonar, Bus CAN, etc.) and robotic platforms (Jaguar, VIPALAB, RobuFAST, Camaleon, etc.). Moreover, a recorded data allows a real-time replaying according to data acquisition from sensors. This recorded data can be exported to scientific softwares such as MATLAB®.

G.3/ IMPLEMENTATION OF THE CONTROL ARCHITECTURE

The proposed control architecture and each functionality (perception, localization, control, etc.) for autonomous navigation has been implemented using C++. A class diagram summarizing the main programming functions and the interaction between them is given in Fig. G.3.

In these experiments, each vehicle uses a combination of RTK-GPS and IMU using EKF to estimate its localization (current position and orientation) at a sample time of $T_s = 0.1\ s$ (cf. Fig. G.3).

These sensors provide enough accurate data w.r.t. the vehicle dynamic. Indeed, in these experiments, the vehicles move at maximum velocity of $2.5\ m/s$ due to the short dimensions of the used urban platform (cf. Fig. G.2). The LIDAR is a very useful sensor to detect obstacles, nevertheless, according to the environment (e.g., PAVIN), the LIDAR can detect the floor and white lines on it. This detection of floor can generate an incorrect application of obstacle avoidance. To avoid this situation, the LIDAR data was filtered

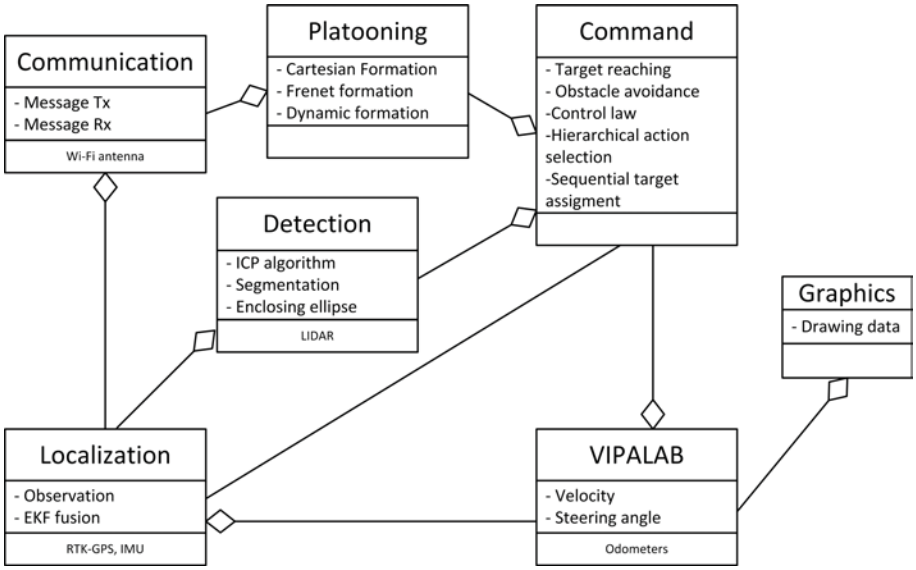


Figure G.3: Class diagram of the proposed control architecture implementation in each VIPALAB.

according to the altitude w.r.t. the UGV. The LIDAR perception is processed by the detection method to obtain the obstacles' features. Moreover, each VIPALAB can communicate via WI-FI. It allows to the leader to send its current configuration (pose and velocity) to other vehicles (cf. Chapter 5). The delay in the communication system is dealt with an EKF to obtain information at each sample time ($T_s = 0.1\ s$). The proposed method for each behaviors and the navigation strategy are implemented in Command and platooning classes.

LIST OF MY PUBLICATIONS

[1] VILCA, J., ADOUANE, L., AND MEZOUAR, Y. **On-line obstacle detection using data range for reactive obstacle avoidance**. In *12th International Conference on Intelligent Autonomous Systems* (June 2012), Korea.

[2] VILCA, J., ADOUANE, L., AND MEZOUAR, Y. **Robust on-line obstacle detection using range data for reactive navigation**. In *10th International IFAC Symposium on Robot Control* (September 2012), Croatia.

[3] VILCA, J., ADOUANE, L., AND MEZOUAR, Y. **Reactive navigation of a mobile robot using elliptic trajectories and effective online obstacle detection**. *Gyroscopy and Navigation - Springer Verlag 4* (2013), 14 – 25.

[4] VILCA, J., ADOUANE, L., MEZOUAR, Y., AND LÉBRALY, P. **An overall control strategy based on target reaching for the navigation of an urban electric vehicle**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13)* (Tokyo - Japan, November 2013).

[5] VILCA, J. M., ADOUANE, L., AND MEZOUAR, Y. **Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration**. In *Springer Tracts in Advanced Robotics, from International Symposium on Distributed Autonomous Robotic Systems, DARS 2014* (Daejeon - Korea, 2-5 November 2014), pp. 727–734.

[6] DAFFLON, B., VILCA, J., GECHTER, F., AND ADOUANE, L. **Adaptive autonomous navigation using reactive multi-agent system for control law merging**. In *International Conference on Computational Science (ICCS)* (Iceland, June 2015).

[7] VILCA, J., ADOUANE, L., AND MEZOUAR, Y. **Constrained leader-follower navigation of vehicles using adaptive dynamic of the leader (submitted)**. *Advanced Robotics* (2015).

[8] VILCA, J. M., ADOUANE, L., AND MEZOUAR, Y. **A novel safe and flexible control strategy based on target reaching for the navigation of urban vehicles.** *Robotics and Autonomous Systems journal* (2015).

[9] VILCA, J. M., ADOUANE, L., AND MEZOUAR, Y. **Optimal multi-criteria waypoint selection for autonomous vehicle navigation in structured environment**. *Intelligent and Robotic Systems journal* (2015).

BIBLIOGRAPHY

[10] ČAPEK, K. **Rosumovi univerzálni roboti (r.u.r)**, 1920.

[11] ASIMOV, I. **Liar!** Astounding Science Fiction, 1941.

[12] WALTER, W. G. **The Living Brain**. W. W. Norton, New York, 1953.

[13] BELLMAN, R. **A markovian decision process**. *Journal of Mathematics and Mechanics 6* (1957), 679 – 684.

[14] NILSSON, N. J. **A mobile automaton: An application of ai techniques**. In *Proc. of the First International Joint Conference on Artificial Intelligence* (1969), Morgan Kaufmann Publishers, pp. 509 – 520.

[15] NILSSON, N. J. **Principles of Artificial Intelligence**. Springer, 1982.

[16] HAYES-ROTH, B. **A blackboard architecture for control**. *Artif. Intell. 26*, 3 (August 1985), 251–321.

[17] BROOKS, R. A. **A robust layered control system for a mobile robot**. *IEEE Journal of Robotics and Automation RA-2* (March 1986), pp.14–23.

[18] KHATIB, O. **Real-time obstacle avoidance for manipulators and mobile robots**. *The International Journal of Robotics Research 5* (Spring 1986), pp.90–99.

[19] PAYTON, D. **An architecture for reflexive autonomous vehicle control**. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on* (Apr 1986), vol. 3, pp. 1838–1845.

[20] ELFES, A. **Sonar-based real-world mapping and navigation**. *Robotics and Automation, IEEE Journal of 3*, 3 (June 1987), 249–265.

[21] NAKAMURA, Y., HANAFUSA, H., AND YOSHIKAWA, T. **Task-priority based redundancy control of robot manipulators**. *International Journal of Robotics Researach 6*, 2 (July 1987), 3–15.

[22] BENI, G. **The concept of cellular robotic system**. In *Intelligent Control, 1988. Proceedings., IEEE International Symposium on* (Aug 1988), pp. 57–62.

[23] FUKUDA, T., AND NAKAGAWA, S. **Approach to the dynamically reconfigurable robotic system**. *Journal of Intelligent and Robotic Systems 1*, 1 (1988), 55–72.

[24] FUKUDA, T., AND NAKAGAWA, S. **Dynamically reconfigurable robotic system**. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on* (Apr 1988), pp. 1581–1586 vol.3.

[25] THOMAS, G., AND FINNEY, R. **Calculus and Analytic Geometry**, 7 ed. Addison-Wesley, 1988.

[26] WILLIAMS, M. **Prometheus-the european research programme for optimising the road transport system in europe**. In *Driver Information, IEE Colloquium on* (Dec 1988), pp. 1/1–1/9.

[27] ALBUS, J., MCCAIN, H., AND LUMIA, R. **Nasa/nbs standard reference model for telerobot control system architecture (nasrem)**. Tech. Rep. 1235, National Institute of Stadarnds and Technology, 1989.

[28] ARAI, T., OGATA, H., AND SUZUKI, T. **Collision avoidance among multiple robots using virtual impedance**. In *Intelligent Robots and Systems' 89. The Autonomous Mobile Robots and Its Applications. IROS'89. Proceedings., IEEE/RSJ International Workshop on* (1989), IEEE, pp. 479–485.

[29] ARKIN, R. C. **Motor schema-based mobile robot navigation**. *International Journal of Robotics Research 8*, 4 (1989), pp.92–112.

[30] ASAMA, H., MATSUMOTO, A., AND ISHIDA, Y. **Design of an autonomous and distributed robot system: Actress**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (1989), pp. 283–290.

[31] KENDALL, D. G. **A survey of the statistical theory of shape**. *Statistical Science 4*, 2 (May 1989), 87 – 99.

[32] STEELS, L. **Cooperation between distributed agents through self-orcamsation**.

[33] WANG, P. **Navigation strategies for multiple autonomous mobile robots moving in formation**. In *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on* (Sep 1989), pp. 486–493.

[34] ASAMA, H. **A concept of distributed architecture for maintenance robot systems**. In *Proceedings of the first international conference on supercomputing in nuclear applications (SNA'90)* (1990).

[35] CALOUD, P., CHOI, W., LATOMBE, J.-C., LE PAPE, C., AND YIM, M. **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* (Jul 1990), pp. 67–72 vol.1.

[36] HELLÅKER, J. **Prometheus-strategy**. Tech. rep., SAE Technical Paper, 1990.

[37] KANAYAMA, Y., KIMURA, Y., MIYAZAKI, F., AND NOGUCHI, T. **A stable tracking control method for an autonomous mobile robot**. In *Proceedings of the IEEE International Conference on Robotics and Automation* (May 1990), pp. 384 – 389.

[38] LE PAPE, C. **A combination of centralized and distributed methods for multi-agent planning and scheduling**. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on* (1990), IEEE, pp. 488–493.

[39] PREMVUTI, S., AND YUTA, S. **Consideration on the cooperation of multiple autonomous mobile robots**. In *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on* (Jul 1990), pp. 59–63 vol.1.

[40] SUGIHARA, K., AND SUZUKI, I. **Distributed motion coordination of multiple mobile robots**. In *Intelligent Control, 1990. Proceedings., 5th IEEE International Symposium on* (1990), IEEE, pp. 138–143.

[41] ALBUS, J. **Outline of a theory of intelligence**. *IEEE Transaction on Systems, Man and CIbernetics 21*, 3 (May 1991), 473–509.

[42] BARSHAN, B., AND KUC, R. **Active sonar for obstacle localization using envelope shape information**. In *International Conference on Acoustics, Speech, and Signal Processing* (April 1991), vol. 2, pp. 1273 – 1276.

[43] BROOKS, R. A. **New approaches to robotics**. *Science 253*, 5025 (1991), 1227–1232.

[44] DICKMANNS, E. **4-d dynamic vision for intelligent motion control**. *Engineering Applications of Artificial Intelligence 4*, 4 (1991), 301 – 307.

[45] KOREN, Y., AND BORENSTEIN, J. **Potential field methods and their inherent limitations for mobile robot navigation**. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (Apr 1991), pp. 1398–1404 vol.2.

[46] LATOMBE, J.-C. **Robot Motion Planning**. Kluwer Academic Publishers, Boston, MA, 1991.

[47] OLIN, K., AND TSENG, D. **Autonomous cross-country navigation: an integrated perception and planning system**. *IEEE Expert 6*, 4 (Aug 1991), 16–30.

[48] SHLADOVER, S., DESOER, C., HEDRICK, J., TOMIZUKA, M., WALRAND, J., ZHANG, W.-B., MCMAHON, D., PENG, H., SHEIKHOLESLAM, S., AND MCKEOWN, N. **Automated vehicle control developments in the path program**. *Vehicular Technology, IEEE Transactions on 40*, 1 (Feb 1991), 114–130.

[49] SHLADOVER, S. E., DESOER, C. A., HEDRICK, J. K., TOMIZUKA, M., WALRAND, J., ZHANG, W.-B., MCMAHON, D. H., PENG, H., SHEIKHOLESLAM, S., AND MCKEOWN, N. **Automated vehicle control developments in the path program**. *Vehicular Technology, IEEE Transactions on 40*, 1 (1991), 114–130.

[50] WANG, P. **Navigation strategies for multiple autonomous mobile robots moving in formation**. *Journal of Robotic Systems 8*, 2 (1991), 177–195.

[51] WELZL, E. **Smallest enclosing disks (balls and ellipsoids)**. In *Results and New Trends in Computer Science* (1991), Springer-Verlag, pp. 359–370.

[52] RIMON, E., AND DANIEL.KODITSCHEK. **Exact robot navigation using artificial potential fields**. *IEEE Transactions on Robotics and Automation 8*, 5 (1992), 501 – 518.

[53] SIMMONS, R. **Monitoring and error recovery for autonomous walking**. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on* (Jul 1992), vol. 2, pp. 1407–1412.

[54] YUTA, S., AND PREMVUTI, S. **Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots**. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on* (1992), vol. 3, IEEE, pp. 1566–1574.

[55] BENI, G., AND WANG, J. **Swarm intelligence in cellular robotic systems**. In *Robots and Biological Systems: Towards a New Bionics?*, P. Dario, G. Sandini, and P. Aebischer, Eds., vol. 102 of *NATO ASI Series*. Springer Berlin Heidelberg, 1993, pp. 703–712.

[56] MATARIC, M. J. **Designing emergent behaviors: From local interactions to collective intelligence**. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior* (1993), pp. 432–441.

[57] MICHALSKA, H., AND MAYNE, D. Q. **Robust receding horizon control of constrained nonlinear systems**. *Automatic Control, IEEE Transactions on 38*, 11 (1993), 1623–1633.

[58] NOREILS, F. R. **Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment**. *The International Journal of Robotics Research 12*, 1 (1993), 79–98.

[59] PARKER, L. E. **Designing control laws for cooperative agent teams**. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* (1993), IEEE, pp. 582–587.

[60] UEYAMA, T., AND FUKUDA, T. **Self-organization of cellular robots using random walk with simple rules**. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* (1993), IEEE, pp. 595–600.

[61] CHEN, Q., AND LUH, J. **Distributed motion coordination of multiple robots**. In *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on* (Sep 1994), vol. 3, pp. 1493–1500 vol.3.

[62] DE LUCA, A., AND ORIOLO, G. **Local incremental planning for nonholonomic mobile robots**. In *IEEE International Conference on Robotics and Automation* (May 1994), vol. 1, pp. 104 – 110.

[63] DICKMANNS, E., BEHRINGER, R., DICKMANNS, D., HILDEBRANDT, T., MAURER, M., THOMANEK, F., AND SCHIEHLEN, J. **The seeing passenger car 'vamors-p'**. In *Intelligent Vehicles '94 Symposium, Proceedings of the* (Oct 1994), pp. 68–73.

[64] JIN, K., LIANG, P., AND BENI, G. **Stability of synchronized distributed control of discrete swarm structures**. In *Proceedings in IEEE International Conference on Robotics and Automation* (1994), IEEE, pp. 1033–1038.

[65] LEE, J., HUBER, M. J., DURFEE, E. H., AND KENNY, P. G. **Um-prs: An implementation of the procedural reasoning system for multirobot applications**. *Conference on Intelligent Robotics in Field, Factory, Service and Space (CIRFFSS) 2* (1994), 842–849.

[66] PARKER, L. E. **Alliance: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots**. In *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on* (1994), vol. 2, IEEE, pp. 776–783.

[67] SIMMONS, R. G. **Structured control for autonomous robots**. *IEEE Transaction on Robotics and Automation 10*, 1 (1994), 34–43.

[68] ULMER, B. **Vita ii-active collision avoidance in real traffic**. In *Intelligent Vehicles '94 Symposium, Proceedings of the* (Oct 1994), pp. 1–6.

[69] AICARDI, M., CASALINO, G., BICCHI, A., AND BALESTRINO, A. **Closed loop steering of unicycle like vehicles via lyapunov techniques**. *Robotics Automation Magazine, IEEE 2*, 1 (1995), 27–35.

[70] BARSHAN, B., AND DURRANT-WHYTE, H. **Evaluation of a solid-state gyroscope for robotics applications**. *Instrumentation and Measurement, IEEE Transactions on 44*, 1 (Feb 1995), 61–67.

[71] BERTSEKAS, D. P. **Dynamic Programming and Optimal Control**, vol. I. Belmont Massachusetts - Athena Scientific, 1995.

[72] CAI, A.-H., FUKUDA, T., ARAI, F., UEYAMA, T., AND SAKAI, A. **Hierarchical control architecture for cellular robotic system-simulations and experiments**. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on* (May 1995), vol. 1, pp. 1191–1196 vol.1.

[73] CHATILA, R. **Deliberation and reactivity in autonomous mobile robots**. *Robotics and Autonomous Systems 16* (1995), 197–211.

[74] IRITANI, G. **Construction mechanism of group behavior with cooperation**. In *Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3-Volume 3* (1995), IEEE Computer Society, p. 3535.

[75] MATARIC, M., NILSSON, M., AND SIMSARIN, K. **Cooperative multi-robot box-pushing**. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on* (Aug 1995), vol. 3, pp. 556–561 vol.3.

[76] MATARIĆ, M. J. **Issues and approaches in the design of collective autonomous agents**. *Robotics and autonomous systems 16*, 2 (1995), 321–331.

[77] SAMSON, C. **Control of chained systems. application to path following and time-varying point-stabilization of mobile robots**. *IEEE Transactions on Automatic Control 40*, 1 (January 1995), 64–77.

[78] SCHÖNER, G., DOSE, M., AND ENGELS, C. **Dynamics of behavior: Theory and applications for autonomous robot architectures**. *Robotics and Autonomous Systems 16*, 2âĂŞ4 (1995), 213 – 245. Moving the Frontiers between Robotics and Biology.

[79] TSOTSOS, J. K. **Behaviorist intelligence and the scaling problem**. *Artif. Intell. 75*, 2 (June 1995), 135–160.

[80] WEBSITE, E. **http://www.eurekanetwork.org/project/-/id/45**, January 1995.

[81] KAMON, I., RIVLIN, E., AND RIMON, E. **A new range-sensor based globally convergent navigation algorithm for mobile robots**. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* (Apr 1996), vol. 1, pp. 429–435 vol.1.

[82] LOAN, C. F. V. **Introduction to Computational Science and Mathematics**. Jones and Bartlett Publishers„ 1996.

[83] MATTHIES, L., KELLY, A., LITWIN, T., AND THARP, G. **Obstacle detection for unmanned ground vehicles: A progress report**. In *Robotics Research*, G. Giralt and G. Hirzinger, Eds. Springer London, 1996, pp. 475–486.

[84] O'HARE, G. M., AND JENNINGS, N. **Foundations of distributed artificial intelligence**, vol. 9. John Wiley & Sons, 1996.

[85] PARKER, L. E. **L-alliance: Task-oriented multi-robot learning in behavior-based systems**. *Advanced Robotics 11*, 4 (1996), 305–322.

[86] TAN, K.-H., AND LEWIS, M. A. **Virtual structures for high-precision cooperative mobile robotic control**. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on* (1996), vol. 1, IEEE, pp. 132–139.

[87] THOMANEK, F., AND DICKMANNS, E. **Autonomous road vehicle guidance in normal traffic**. In *Recent Developments in Computer Vision*, S. Li, D. Mital, E. Teoh, and H. Wang, Eds., vol. 1035 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1996, pp. 499–507.

[88] CAO, Y. U., FUKUNAGA, A. S., AND KAHNG, A. B. **Cooperative mobile robotics: Antecedents and directions**. *Autonomous Robots 4* (1997), 226–234.

[89] DAVIET, P., AND PARENT, M. **Platooning for small public urban vehicles**. In *Experimental Robotics IV*, O. Khatib and J. Salisbury, Eds., vol. 223 of *Lecture Notes in Control and Information Sciences*. Springer Berlin Heidelberg, 1997, pp. 343–354.

[90] HOLLAND, O. **Grey walter: The pioneer of real artificial life**. In *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems* (1997), pp. 34 – 41.

[91] KHEDDAR, A., TZAFESTAS, C., COIFFET, P., KOTOKU, T., KAWABATA, S., IWAMOTO, K., TANIE, K., MAZON, I., LAUGIER, C., AND CHELLALI, R. **Parallel multi-robots long distance teleoperation**. In *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on* (Jul 1997), pp. 1007–1012.

[92] LEWIS, M. A., AND TAN, K. H. **High precision formation control of mobile robots using virtual structures**. *Autonomous Robots 4*, 4 (1997), 387 – 403.

[93] MATIJEVIC, J. **Sojourner: The mars pathfinder microrover flight experiment**. *Space Technology 17*, 3 (1997), 143–149.

[94] MATIJEVIC, J., AND SHIRLEY, D. **The mission and operation of the mars pathfinder microrover**. *Control Engineering Practice 5*, 6 (1997), 827–835.

[95] MAURER, M., AND DICKMANNS, E. **A system architecture for autonomous visual road vehicle guidance**. In *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on* (Nov 1997), pp. 578–583.

[96] MEDEIROS, A., AND CHATILA, R. **Priorities and data abstraction in hierarchical control architecture for autonomous robots**. In *Proc. of Workshop on Intelligent Robotics* (1997).

[97] ZHANG, Z. **Parameter estimation techniques: A tutorial with application to conic fitting**. *Image and Vision Computing 15* (1997), 59 – 76.

[98] ALAMI, R., CHATILA, R., FLEURY, S., GHALLAB, M., AND INGRAND, F. **An architecture for autonomy**. *The International Journal of Robotics Research 17*, 4 (1998), 315–337.

[99] ALAMI, R., FLEURY, S., HERRB, M., INGRAND, F., AND ROBERT, F. **Multi-robot cooperation in the martha project**. *Robotics & Automation Magazine, IEEE 5*, 1 (1998), 36–47.

[100] ARKIN, R. C. **Behavior-Based Robotics**. The MIT Press, 1998.

[101] BALCH, T., AND ARKIN, R. **Behavior-based formation control for multirobot teams**. *Robotics and Automation, IEEE Transactions on 14*, 6 (1998), 926–939.

[102] BEHRINGER, R., AND MULLER, N. **Autonomous road vehicle guidance from autobahnen to narrow curves**. *Robotics and Automation, IEEE Transactions on 14*, 5 (Oct 1998), 810–815.

[103] COOPER, B. **Driving on the surface of mars using the rover control workstation**. *Proceedings of SpaceOps 1998* (1998).

[104] DESAI, J., OSTROWSKI, J., AND KUMAR, V. **Controlling formations of multiple mobile robots**. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on* (May 1998), vol. 4, pp. 2864–2869 vol.4.

[105] GUTMANN, J.-S., BURGARD, W., FOX, D., AND KONOLIGE, K. **An experimental comparison of localization methods**. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on* (Oct 1998), vol. 2, pp. 736–743 vol.2.

[106] KHOSHNEVIS, B., AND BEKEY, G. **Centralized sensing and control of multiple mobile robots**. *Computers & Industrial Engineering 35*, 3âĂŞ4 (1998), 503 – 506. Selected Papers from the 22nd {ICC} and {IE} Conference.

[107] LUCA, A. D., ORIOLO, G., AND SAMSON, C. **Feedback control of a nonholonomic car-like robot**. In *Robot Motion Planning and Control*, J.-P. Laumond, Ed. Springer-Verlag, Berlin, 1998, pp. 171–253.

[108] PARKER, L. E. **Alliance: An architecture for fault tolerant multirobot cooperation**. *IEEE Transactions on Robotics and Automation 14*, 2 (1998), 220–240.

[109] RAJAMANI, R., CHOI, S., HEDRICK, J., AND LAW, B. **Design and experimental implementation of control for a platoon of automated vehicles**. In *Proceedings of the ASME Dynamic Systems and Control Division (1998)* (1998).

[110] TAN, H.-S., RAJAMANI, R., AND ZHANG, W.-B. **Demonstration of an automated highway platoon system**. In *American Control Conference, 1998. Proceedings of the 1998* (Jun 1998), vol. 3, pp. 1823–1827 vol.3.

[111] FRAICHARD, T. **Trajectory planning in a dynamic workspace: a "state time" approach**. *Advanced Robotics 13*, 1 (1999), 75–94.

[112] KITANO, H., TADOKORO, S., NODA, I., MATSUBARA, H., TAKAHASHI, T., SHINJOU, A., AND SHIMADA, S. **Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research**. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on* (1999), vol. 6, pp. 739–743 vol.6.

[113] MESBAHI, M., AND HADAEGH, F. **Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching**. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on* (1999), vol. 2, pp. 1211–1216 vol. 2.

[114] SCOKAERT, P. O., MAYNE, D. Q., AND RAWLINGS, J. B. **Suboptimal model predictive control (feasibility implies stability)**. *Automatic Control, IEEE Transactions on 44*, 3 (1999), 648–654.

[115] VOLPE, R. **Navigation results from desert field tests of the rocky 7 mars rover prototype**. *The International Journal of Robotics Research 18*, 7 (1999), 669–683.

[116] BEARD, R. W., LAWTON, J., AND HADAEGH, F. Y. **A feedback architecture for formation control**. In *Proceedings of the American Control Conference* (2000), vol. 6, pp. 4087–4091.

[117] BULUSU, N., HEIDEMANN, J., AND ESTRIN, D. **Gps-less low-cost outdoor localization for very small devices**. *Personal Communications, IEEE 7*, 5 (Oct 2000), 28–34.

[118] CHIEN, S., BARRETT, A., ESTLIN, T., AND RABIDEAU, G. **A comparison of coordinated planning methods for cooperating rovers**. In *Proceedings of the Fourth International Conference on Autonomous Agents* (New York, NY, USA, 2000), AGENTS '00, ACM, pp. 100–101.

[119] DE MAESSCHALCK, R., JOUAN-RIMBAUD, D., AND MASSART, D. **The mahalanobis distance**. *Chemometrics and Intelligent Laboratory Systems 50*, 1 (2000), 1 – 18.

[120] LEE, J. **Reactive-system approaches to agent architectures**. In *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, N. Jennings and Y. LespÃľrance, Eds., vol. 1757 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2000, pp. 132–146.

[121] MAYNE, D. Q., RAWLINGS, J. B., RAO, C. V., AND SCOKAERT, P. O. **Constrained model predictive control: Stability and optimality**. *Automatica 36*, 6 (2000), 789–814.

[122] PARKER, L. **Current state of the art in distributed autonomous mobile robotics**. In *Distributed Autonomous Robotic Systems 4*, L. Parker, G. Bekey, and J. Barhen, Eds. Springer Japan, 2000, pp. 3–12.

[123] SZCZERBA, R., GALKOWSKI, P., GLICKTEIN, I., AND TERNULLO, N. **Robust algorithm for real-time route planning**. *IEEE Transactions on Aerospace and Electronic Systems 36*, 3 (2000), 869–878.

[124] BEARD, R., LAWTON, J., AND HADAEGH, F. **A coordination architecture for spacecraft formation control**. *IEEE Transactions on Control Systems Technology 9* (2001), 777 – 790.

[125] CHANDLER, P. R., PACHTER, M., AND RASMUSSEN, S. **Uav cooperative control**. In *American Control Conference, 2001. Proceedings of the 2001* (2001), vol. 1, pp. 50–55 vol.1.

[126] DÉFAGO, X. **Distributed computing on the move: From mobile computing to cooperative robotics and nanorobotics**. In *Proc. 1st Wksp. on Principles of Mobile Computing (POMC)* (2001).

[127] DESAI, J., OSTROWSKI, J., AND KUMAR, V. **Modeling and control of formations of nonholonomic mobile robots**. *IEEE Transaction on Robotics and Automation 17*, 6 (2001), 905 – 908.

[128] FIERRO, R., DAS, A., KUMAR, V., AND OSTROWSKI, J. **Hybrid control of formations of robots**. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 1, pp. 157–162 vol.1.

[129] HEALEY, A. **Application of formation control for multi-vehicle robotic minesweeping**. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on* (2001), vol. 2, pp. 1497–1502 vol.2.

[130] HOFMANN, U., RIEDER, A., AND DICKMANNS, E. **Radar and vision data fusion for hybrid adaptive cruise control on highways**. In *Computer Vision Systems*, B. Schiele and G. Sagerer, Eds., vol. 2095 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2001, pp. 125–138.

[131] INGRAND, F., CHATILA, R., AND ALAMI, R. **An architecture for dependable autonomous robots.** In *ETFA (2)* (2001), pp. 657–658.

[132] LAUMOND, J.-P. **La Robotique Mobile, Chapitre 2**. Traité IC2 Information-Commande-Communication. Hermès, 2001.

[133] LEONARD, N. E., AND FIORELLI, E. **Virtual leaders, artificial potentials and coordinated control of groups**. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on* (2001), vol. 3, IEEE, pp. 2968–2973.

[134] OGREN, P., EGERSTEDT, M., AND HU, X. **A control lyapunov function approach to multi-agent coordination**. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on* (2001), vol. 2, IEEE, pp. 1150–1155.

[135] PIRJANIAN, P., HUNTSBERGER, T. L., AND SCHENKER, P. S. **Development of campout and its further applications to planetary rover operations: A multirobot control architecture**. In *Intelligent Systems and Advanced Manufacturing* (2001), International Society for Optics and Photonics, pp. 108–119.

[136] SIMMONS, R., SINGH, S., HERSHBERGER, D., RAMOS, J., AND SMITH, T. **First results in the coordination of heterogeneous robots for large-scale assembly**. In *Experimental Robotics VII*. Springer, 2001, pp. 323–332.

[137] SPARBERT, J., DIETMAYER, K., AND STRELLER, D. **Lane detection and street type classification using laser range images**. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE* (2001), pp. 454–459.

[138] THRAPP, R., WESTBROOK, C., AND SUBRAMANIAN, D. **Robust localization algorithms for an autonomous campus tour guide**. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 2, pp. 2065–2071 vol.2.

[139] VOLPE, R., NESNAS, I., ESTLIN, T., MUTZ, D., PETRAS, R., AND DAS, H. **The claraty architecture for robotic autonomy**. In *Aerospace Conference, 2001, IEEE Proceedings.* (2001), vol. 1, pp. 1/121–1/132 vol.1.

[140] CARPIN, S., AND PAGELLO, E. **On parallel rrts for multi-robot systems**. In *Proc. 8th Conf. Italian Association for Artificial Intelligence* (2002), pp. 834–841.

[141] CORTES, J., MARTINEZ, S., KARATAS, T., AND BULLO, F. **Coverage control for mobile sensing networks**. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (2002), vol. 2, IEEE, pp. 1327–1332.

[142] DAS, A., FIERRO, R., KUMAR, V., OSTROWSKI, J., SPLETZER, J., AND TAYLOR, C. **A vision-based formation control framework**. *IEEE Transaction on Robotics and Automation 18*, 5 (2002), 813–825.

[143] EGERSTEDT, M., AND HU, X. **A hybrid control approach to action coordination for mobile robots**. *Automatica 38(1)* (2002), 125–130.

[144] FEDDEMA, J., LEWIS, C., AND SCHOENWALD, D. **Decentralized control of cooperative robotic vehicles: theory and application**. *Robotics and Automation, IEEE Transactions on 18*, 5 (Oct 2002), 852–864.

[145] FIERRO, R., DAS, A., SPLETZER, J., ESPOSITO, J., KUMAR, V., OSTROWSKI, J. P., PAPPAS, G., TAYLOR, C. J., HUR, Y., ALUR, R., LEE, I., GRUDIC, G., AND SOUTHALL, B. **A framework and architecture for multi-robot coordination**. *The International Journal of Robotics Research 21*, 10-11 (2002), 977–995.

[146] FREDSLUND, J., AND MATARIC, M. J. **A general algorithm for robot formations using local sensing and minimal communication**. *Robotics and Automation, IEEE Transactions on 18*, 5 (2002), 837–846.

[147] KHALIL, H. K. **Nonlinear Systems**, 3 ed. Prentice Hall, 2002.

[148] KIM, J., PEARCE, R., AND AMATO, N. **Robust geometric-based localization in indoor environments using sonar range sensors**. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on* (2002), vol. 1, pp. 421–426 vol.1.

[149] LIU, Y., AND PASSINO, K. **Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors**. *Journal of optimization theory and applications 115*, 3 (2002), 603–628.

[150] MURPHY, R. R., LISETTI, C. L., TARDIF, R., IRISH, L., AND GAGE, A. **Emotion-based control of cooperating heterogeneous mobile robots**. *Robotics and Automation, IEEE Transactions on 18*, 5 (2002), 744–757.

[151] NAGPAL, R. **Programmable self-assembly using biologically-inspired multiagent control**. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1* (2002), ACM, pp. 418–425.

[152] NICOLESCU, M. N., AND MATARIĆ, M. J. **A hierarchical architecture for behavior-based robots**. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1* (2002), ACM, pp. 227–233.

[153] OLFATI-SABER, R., AND MURRAY, R. M. **Distributed cooperative control of multiple vehicle formations using structural potential functions**. In *IFAC World Congress* (2002), Citeseer, pp. 346–352.

[154] SERAJI, H., AND HOWARD, A. **Behavior-based robot navigation on challenging terrain: A fuzzy logic approach**. *Robotics and Automation, IEEE Transactions on 18*, 3 (Jun 2002), 308–321.

[155] STOETER, S. A., RYBSKI, P. E., STUBBS, K. N., MCMILLEN, C. P., GINI, M., HOUGEN, D. F., AND PAPANIKOLOPOULOS, N. **A robot team for surveillance tasks: Design and architecture**. *Robotics and Autonomous Systems 40*, 2-3 (2002), 173 – 183.

[156] VIDAL, R., SHAKERNIA, O., KIM, H. J., SHIM, H., AND SASTRY, S. **Multi-agent probabilistic pursuit-evasion games with unmanned ground and aerial vehicles**. *IEEE Transactions on Robotics and Automation 18*, 5 (2002), 662–669.

[157] BARAS, J., TAN, X., AND HOVARESHTI, P. **Decentralized control of autonomous vehicles**. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on* (Dec 2003), vol. 2, pp. 1532–1537 Vol.2.

[158] DAS, A., FIERRO, R., AND KUMAR, V. **Control graphs for robot networks**. In *Cooperative Control: Models, Applications and Algorithms*. Springer, 2003, pp. 55–73.

[159] DIMAROGONAS, D., ZAVLANOS, M., LOIZOU, S., AND KYRIAKOPOULOS, K. **Decentralized motion control of multiple holonomic agents under input constraints**. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on* (Dec 2003), vol. 4, pp. 3390–3395 vol.4.

[160] DOH, N., CHOSET, H., AND CHUNG, W. K. **Accurate relative localization using odometry**. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* (Sept 2003), vol. 2, pp. 1606–1612 vol.2.

[161] HUNTSBERGER, T., PIRJANIAN, P., TREBI-OLLENNU, A., DAS NAYAR, H., AGHAZARIAN, H., GANINO, A. J., GARRETT, M., JOSHI, S. S., AND SCHENKER, P. S. **Campout: A control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration**. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 33*, 5 (2003), 550–559.

[162] JADBABAIE, A., LIN, J., AND MORSE, A. S. **Coordination of groups of mobile autonomous agents using nearest neighbor rules**. *Automatic Control, IEEE Transactions on 48*, 6 (2003), 988–1001.

[163] KIM, D.-H., AND KIM, J.-H. **A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer**. *Robotics and Autonomous Systems 42(1)* (2003), 17–30.

[164] LAWTON, J., BEARD, R., AND YOUNG, B. **A decentralized approach to formation maneuvers**. *Robotics and Automation, IEEE Transactions on 19*, 6 (2003), 933–941.

[165] LIAN, F.-L., AND MURRAY, R. **Cooperative task planning of multi-robot systems with temporal constraints**. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* (Sept 2003), vol. 2, pp. 2504–2509.

[166] NESNAS, I. A., WRIGHT, A., BAJRACHARYA, M., SIMMONS, R., ESTLIN, T., AND KIM, W. S. **Claraty: An architecture for reusable robotic software**. In *AeroSense 2003* (2003), International Society for Optics and Photonics, pp. 253–264.

[167] OREBÄCK, A., AND CHRISTENSEN, H. **Evaluation of architectures for mobile robotics**. *Autonomous Robots 14*, 1 (2003), 33–49.

[168] REN, W., BEARD, R. W., AND BEARD, A. W. **Decentralized scheme for spacecraft formation flying via the virtual structure approach**. *AIAA Journal of Guidance, Control, and Dynamics 27* (2003), 73–82.

[169] RUSU, P., PETRIU, E. M., WHALEN, T. E., CORNELL, A., AND SPOELDER, H. J. **Behavior-based neuro-fuzzy controller for mobile robot navigation**. *IEEE Transactions on Instrumentation and Measurement 52*, 4 (2003), 1335–1340.

[170] SCHENKER, P., HUNTSBERGER, T., PIRJANIAN, P., BAUMGARTNER, E., AND TUNSTEL, E. **Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization**. *Autonomous Robots 14*, 2-3 (2003), 103–126.

[171] ZHOU, C., WEI, Y., AND TAN, T. **Mobile robot self-localization based on global visual appearance features**. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* (Sept 2003), vol. 1, pp. 1271–1276 vol.1.

[172] ABUHADROUS, I., AMMOUN, S., NASHASHIBI, F., GOULETTE, F., AND LAURGEAU, C. **Digitizing and 3d modeling of urban environments and roads using vehicle-borne laser scanner system**. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (2004), vol. 1, IEEE, pp. 76–81.

[173] BELTA, C., AND KUMAR, V. **Optimal motion generation for groups of robots: a geometric approach**. *Journal of Mechanical Design 126*, 1 (2004), 63–70.

[174] BERTUCCELLI, L., ALIGHANBARI, M., AND HOW, J. **Robust planning for coupled cooperative uav missions**. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on* (Dec 2004), vol. 3, pp. 2917–2922 Vol.3.

[175] CHAIMOWICZ, L., KUMAR, R. V., AND CAMPOS, M. F. M. **A mechanism for dynamic coordination of multiple robots**. *Autonomous Robots 17* (2004), 7 – 21.

[176] DUNBAR, W. B., AND MURRAY, R. M. **Receding horizon control of multi-vehicle formations: A distributed implementation**. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on* (2004), vol. 2, IEEE, pp. 1995–2002.

[177] LEMAY, M., MICHAUD, F., LETOURNEAU, D., AND VALIN, J.-M. **Autonomous initialization of robot formations**. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* (2004), vol. 3, pp. 3018–3023 Vol.3.

[178] LENAIN, R., THUILOT, B., CARIOU, C., AND MARTINET, P. **A new nonlinear control for vehicle in sliding conditions: Application to automatic guidance of farm vehicles using RTK GPS**. In *IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans (USA), April 2004), pp. 4381–4386.

[179] OGREN, P., FIORELLI, E., AND LEONARD, N. E. **Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment**. *Automatic Control, IEEE Transactions on 49*, 8 (2004), 1292–1302.

[180] RYAN, A., ZENNARO, M., HOWELL, A., SENGUPTA, R., AND HEDRICK, J. **An overview of emerging results in cooperative uav control**. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on* (Dec 2004), vol. 1, pp. 602–607 Vol.1.

[181] SAFFIOTTI, A. **Platforms for rescue operations**. *AASS Mobile Robotics Laboratory. Orebro University, Orebro, Swed* (2004).

[182] SIEGWART, R., AND NOURBAKHSH, I. R. **Introduction to Autonomous Mobile Robots**. Bradford Company, Scituate, MA, USA, 2004.

[183] SIOURIS, G. M. **Missile Guidance and Control Systems**. Springer-Verlag, 2004.

[184] STANSBURY, R. S., AKERS, E. L., HARMON, H. P., AND AGAH, A. **Survivability, mobility, and functionality of a rover for radars in polar regions**. *International Journal of Control Automation and Systems 2* (2004), 343–353.

[185] TANNER, H., PAPPAS, G., AND KUMAR, V. **Leader-to-formation stability**. *Robotics and Automation, IEEE Transactions on 20*, 3 (June 2004), 443–455.

[186] VOTH, D. **A new generation of military robots**. *Intelligent Systems, IEEE 19*, 4 (Jul 2004), 2–3.

[187] ZAPATA, R., CACITTI, A., AND LEPINAY, P. **Dvz-based collision avoidance control of non-holonomic mobile manipulators**. *JESA, European Journal of Automated Systems 38(5)* (2004), 559–588.

[188] ADOUANE, L. **Architectures de controle comportementales et reactives pour la cooperation d'un groupe de robots mobiles**. PhD thesis, Universite de Franche-Comte, LAB CNRS 6596, April 2005.

[189] ALAMI, R. **Multi-robot cooperation : Architectures and paradigms**. In *Journées nationales de la recherche en robotique (JNRR'05)* (2005).

[190] BURGARD, W., MOORS, M., STACHNISS, C., AND SCHNEIDER, F. **Coordinated multi-robot exploration**. *Robotics, IEEE Transactions on 21*, 3 (June 2005), 376–386.

[191] CHEN, Y. Q., AND WANG, Z. **Formation control: a review and a new consideration**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)* (Aug 2005), pp. 3181–3186.

[192] CHOSET, H., LYNCH, K. M., HUTCHINSON, S., KANTOR, G., BURGARD, W., KAVRAKI, L. E., AND THRUN, S. **Principles of Robot Motion: Theory, Algorithms, and Implementation**. MIT Press, June 2005.

[193] DORIGO, M., AND BLUM, C. **Ant colony optimization theory: A survey**. *Theoretical Computer Science 344*, 2–3 (2005), 243–278.

[194] FANG, H., LENAIN, R., THUILOT, B., AND MARTINET, P. **Trajectory tracking control of farm vehicles in presence of sliding**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005), pp. 58–63.

[195] FRAICHARD, T. **Cybercar: l'alternative à la voiture particulière**. *Navigation (Paris) 53*, 1 (2005), 53–74.

[196] GE, S. S., AND FUA, C.-H. **Queues and artificial potential trenches for multirobot formations**. *Robotics, IEEE Transactions on 21*, 4 (2005), 646–656.

[197] HAZON, N., AND KAMINKA, G. **Redundancy, efficiency and robustness in multi-robot coverage**. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (April 2005), pp. 735–741.

[198] HEGER, F. W., HIATT, L. M., SELLNER, B., SIMMONS, R., AND SINGH, S. **Results in sliding autonomy for multi-robot spatial assembly**. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space* (2005).

[199] JEYARAMAN, S., TSOURDOS, A., ZBIKOWSKI, R., AND WHITE, B. **Formal techniques for the modelling and validation of a co-operating uav team that uses dubins set for path planning**. In *American Control Conference, 2005. Proceedings of the 2005* (June 2005), pp. 4690–4695 vol. 7.

[200] JUR-VAN-DEN, B., AND OVERMARS, M. **Roadmap-based motion planning in dynamic environments**. *IEEE Transactions on Robotics 21(5)* (2005), 885–897.

[201] LINDEMANN, R., AND VOORHEES, C. **Mars exploration rover mobility assembly design, test and performance**. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on* (Oct 2005), vol. 1, pp. 450–455 Vol. 1.

[202] ORTIZ, C. L., VINCENT, R., AND MORISSET, B. **Task inference and distributed task management in the centibots robotic system**. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems* (2005), ACM, pp. 860–867.

[203] PARENT, M., AND DE LA FORTELLE, A. **Cybercars: Past, present and future of the technology**. *arXiv preprint cs/0510059* (2005).

[204] SPLETZER, J. R., AND FIERRO, R. **Optimal positioning strategies for shape changes in robot teams**. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), IEEE, pp. 742–747.

[205] TOMONO, M. **3-d localization and mapping using a single camera based on structure-from-motion with automatic baseline selection**. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (April 2005), pp. 3342–3347.

[206] WENDER, S., SCHOENHERR, M., KAEMPCHEN, N., AND DIETMAYER, K. **Classification of laserscanner measurements at intersection scenarios with automatic parameter optimization**. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE* (2005), IEEE, pp. 94–99.

[207] ADOUANE, L., AND LE FORT-PIAT, N. **Behavioral and distributed control architecture of control for minimalist mobile robots**. *Journal Européen des Systèmes Automatisés 40*, 2 (2006), pp.177–196.

[208] ANTONELLI, G., ARRICHIELLO, F., AND CHIAVERINI, S. **Experiments of formation control with collisions avoidance using the null-space-based behavioral control**. In *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on* (2006), pp. 1–6.

[209] CAMACHO, D., FERNÁNDEZ, F., AND RODELGO, M. A. **Roboskeleton: An architecture for coordinating robot soccer agents**. *Engineering Applications of Artificial Intelligence 19*, 2 (2006), 179 – 188.

[210] CHEN, X., AND LI, Y. **Smooth formation navigation of multiple mobile robots for avoiding moving obstacles**. *International Journal of Control, Automation 4*, 4 (2006), 466 – 479.

[211] CONSOLINI, L., MORBIDI, F., PRATTICHIZZO, D., AND TOSQUES, M. **On the control of a leader-follower formation of nonholonomic mobile robots**. In *Decision and Control, 2006 45th IEEE Conference on* (2006), pp. 5992–5997.

[212] DUNBAR, W. B., AND MURRAY, R. M. **Distributed receding horizon control for multi-vehicle formation stabilization**. *Automatica 42*, 4 (2006), 549–558.

[213] GE, S. S. **Autonomous mobile robots: sensing, control, decision making and applications**. CRC press, 2006.

[214] HORST, J., AND BARBERA, A. **Trajectory generation for an on-road autonomous vehicle**. *Proceedings of the SPIE: Unmanned Systems Technology VIII* (2006).

[215] IAGNEMMA, K., AND BUEHLER, M. **Editorial for journal of field roboticsŮspecial issue on the darpa grand challenge**. *Journal of Field Robotics 23*, 9 (2006), 655–656.

[216] JIE, M. S., BAEK, J. H., HONG, Y. S., AND LEE, K. W. **Real time obstacle avoidance for mobile robot using limit-cycle and vector field method**. *Knowledge-Based Intelligent Information and Engineering Systems* (October 2006).

[217] KOBAYASHI, Y., OTSUBO, K., AND HOSOE, S. **Design of decentralized capturing behavior by multiple mobile robots**. In *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on* (June 2006), pp. 13–18.

[218] KONOLIGE, K., FOX, D., ORTIZ, C., AGNO, A., ERIKSEN, M., LIMKETKAI, B., KO, J., MORISSET, B., SCHULZ, D., STEWART, B., AND OTHERS. **Centibots: Very large scale distributed robotic teams**. In *Experimental Robotics IX*. Springer, 2006, pp. 131–140.

[219] LAMON, P., KOLSKI, S., AND SIEGWART, R. **The smartter-a vehicle for fully autonomous navigation and mapping in outdoor environments**. In *Proceedings of CLAWAR* (2006).

[220] LAMON, P., KOLSKI, S., TRIEBEL, R., SIEGWART, R. Y., BURGARD, W., SIEGWART, R. Y., AND SIEGWART, R. Y. **The smartTer for ELROB 2006-a vehicle for fully autonomous navigation and mapping in outdoor environments**. ETH-Zürich, 2006.

[221] LAVALLE, S. M. **Planning Algorithms**. Cambridge Univ. Press, 2006.

[222] MAALOUF, E., SAAD, M., AND SALIAH, H. **A higher level path tracking controller for a four-wheel differentially steered mobile robot**. *Robotics and Autonomous Systems 54* (2006), 23 – 33.

[223] SINHA, A., AND GHOSE, D. **Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents**. *IEEE Transactions on Automatic Control 51* (2006), 1819–1824.

[224] TANG, H., SONG, A., AND ZHANG, X. **Hybrid behavior coordination mechanism for navigation of reconnaissance robot**. In *International Conference on Intelligent Robots and Systems* (2006).

[225] TRIANNIA, V., NOLFIB, S., AND DORIGOA, M. **Cooperative hole avoidance in a swarm-bot**. *Robotics and Autonomous Systems 54*, 2 (2006), 97–103.

[226] VALDASTRI, P., CORRADI, P., MENCIASSI, A., SCHMICKL, T., CRAILSHEIM, K., SEYFRIED, J., AND DARIO, P. **Micromanipulation, communication and swarm intelligence issues in a swarm microrobotic platform**. *Robotics and Autonomous Systems 54*, 10 (October 2006), 789–804.

[227] ZLOT, R. M., AND STENTZ, A. T. **Market-based multirobot coordination for complex tasks**. *International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics 25*, 1 (January 2006), 73–101.

[228] CLARK, J., AND FIERRO, R. **Mobile robotic sensors for perimeter detection and tracking**. *{ISA} Transactions 46*, 1 (2007), 3 – 13.

[229] CONNORS, J., AND ELKAIM, G. H. **Manipulating b-spline based paths for obstacle avoidance in autonomous ground vehicles**. In *ION National Technical Meeting, ION NTM 2007* (San Diego, CA, USA, 2007).

[230] DIRK, S., AND FRANK, K. **The bio-inspired scorpion robot: design, control & lessons learned**. INTECH Open Access Publisher, 2007.

[231] DO, K., AND PAN, J. **Nonlinear formation control of unicycle-type mobile robots**. *Robotics and Autonomous Systems 55*, 3 (2007), 191 – 204.

[232] GIL PINTO, A., FRAISSE, P., AND ZAPATA, R. **Decentralized strategy for car-like robot formations**. In *International Conference on Intelligent Robots and Systems* (2007).

[233] GOODRICH, M. A., AND SCHULTZ, A. C. **Human-robot interaction: A survey**. *Found. Trends Hum.-Comput. Interact. 1*, 3 (Jan. 2007), 203–275.

[234] LENAIN, R., THUILOT, B., CARIOU, C., AND MARTINET, P. **Adaptive and predictive path tracking control for offroad mobile robots**. *European Journal of Control 13* (2007), 419 – 439.

[235] MACARTHUR, E., AND CRANE, C. **Compliant formation control of a multi-vehicle system**. In *International Symposium on Computational Intelligence in Robotics and Automation - CIRA 2007* (June 2007), pp. 479–484.

[236] MARIOTTINI, G., MORBIDI, F., PRATTICHIZZO, D., PAPPAS, G., AND DANIILIDIS, K. **Leader-follower formations: Uncalibrated vision-based localization and control**. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), pp. 2403–2408.

[237] MINER, D. **Swarm robotics algorithms: A survey**, 2007.

[238] MURRAY, R. M. **Recent research in cooperative control of multivehicle systems**. *J. Dyn. Sys., Meas., Control 129*, 5 (2007), 571–583.

[239] PFEIFER, R., LUNGARELLA, M., AND IIDA, F. **Self-organization, embodiment, and biologically inspired robotics**. *Science Magazine 318*, 5853 (2007), 1088–1093.

[240] PREMEBIDA, C., MONTEIRO, G., NUNES, U., AND PEIXOTO, P. **A lidar and vision-based approach for pedestrian and vehicle detection and tracking**. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE* (Sept 2007), pp. 1044–1049.

[241] THRUN, S., MONTEMERLO, M., DAHLKAMP, H., STAVENS, D., ARON, A., DIEBEL, J., FONG, P., GALE, J., HALPENNY, M., HOFFMANN, G., AND OTHERS. **Stanley: The robot that won the darpa grand challenge**. In *The 2005 DARPA Grand Challenge*. Springer, 2007, pp. 1–43.

[242] TINÓS, R., TERRA, M. H., AND BERGERMAN, M. **A fault tolerance framework for cooperative robotic manipulators**. *Control Engineering Practice 15*, 5 (2007), 615 – 625.

[243] TOIBERO, J., CARELLI, R., AND KUCHEN, B. **Switching control of mobile robots for autonomous navigation in unknown environments**. In *IEEE International Conference on Robotics and Automation* (2007), pp. 1974–1979.

[244] SICILIANO, B., AND KHATIB, O., Eds. **Springer Handbook of Robotics, Part E-34**. Springer, 2008.

[245] BAJRACHARYA, M., MAIMONE, M., AND HELMICK, D. **Autonomy for mars rovers: Past, present, and future**. *Computer 41*, 12 (Dec 2008), 44–50.

[246] BOSSE, M., AND ZLOT, R. **Map matching and data association for large-scale two-dimensional laser scan-based slam**. *The International Journal of Robotics Research 27*, 6 (2008), 667–691.

[247] BOUKERCHE, A., OLIVEIRA, H. A., NAKAMURA, E. F., AND LOUREIRO, A. A. **Vehicular ad hoc networks: A new challenge for localization-based systems**. *Computer Communications 31*, 12 (2008), 2838 – 2849.

[248] CHANG, C.-F., AND FU, L.-C. **A formation control framework based on lyapunov approach**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008), pp. 2777–2782.

[249] CONSOLINI, L., MORBIDI, F., PRATTICHIZZO, D., AND TOSQUES, M. **Leader-follower formation control of nonholonomic mobile robots with input constraints**. *Automatica 44*, 5 (2008), 1343 – 1349.

[250] DJAPIC, V., FARRELL, J., AND DONG, W. **Land vehicle control using a command filtered backstepping approach**. In *American Control Conference, 2008* (June 2008), pp. 2461–2466.

[251] EBERLY, D. **Distance from a point to an ellipse in 2d**. In *Geometric Tools, LLC* (http://www.geometrictools.com/, March 2008).

[252] INOUE, R. S., TERRA, M. H., AND GRASSI JR, V. **Robust state-space estimation for mobile robot localization**. In *Latin American Robotics Symposium and Intelligent Robotics Meeting* (2008), IEEE, pp. 85–90.

[253] KUWATA, Y., FIORE, G. A., TEO, J., FRAZZOLI, E., AND HOW, J. P. **Motion planning for urban driving using rrt**. In *International Conference on Intelligent Robots and Systems* (September 2008), pp. 1681 – 1686.

[254] LABAKHUA, L., NUNES, U., RODRIGUES, R., AND LEITE, F. **Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation**. In *Informatics in Control Automation and Robotics*, J. Cetto, J.-L. Ferrier, J. M. Costa dias Pereira, and J. Filipe, Eds., vol. 15 of *Lecture Notes Electrical Engineering*. Springer Berlin Heidelberg, 2008, pp. 169–182.

[255] PARKER, L. E. **Distributed intelligence: Overview of the field and its application in multi-robot systems**. 5 – 14.

[256] REN, W., AND SORENSEN, N. **Distributed coordination architecture for multi-robot formation control**. *Robotics and Autonomous Systems 56*, 4 (2008), 324 – 333.

[257] TRIVEDI, D., RAHN, C. D., KIER, W. M., AND WALKER, I. D. **Soft robotics: Biological inspiration, state of the art, and future research**. *Applied Bionics and Biomechanics 5*, 3 (2008), 99–117.

[258] WIRASINGHA, S. G., SCHOFIELD, N., AND EMADI, A. **Plug-in hybrid electric vehicle developments in the us: trends, barriers, and economic feasibility**. In *Vehicle Power and Propulsion Conference, 2008. VPPC'08. IEEE* (2008), IEEE, pp. 1–8.

[259] ZIEGLER, J., WERLING, M., AND SCHROEDER, J. **Navigating car-like robots in unstructured environment using an obstacle sensitive cost function**. In *Proc. IEEE Intelligent Vehicle Sympsium (IV)* (Netherlands, 2008), pp. 787 – 791.

[260] ADOUANE, L. **Hybrid and safe control architecture for mobile robot navigation**. In *9th Conference on Autonomous Robot Systems and Competitions* (Portugal, May 2009).

[261] ADOUANE, L. **Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation**. In *9th Conference on Autonomous Robot Systems and Competitions* (Portugal, May 2009).

[262] BERNS, K., AND VON PUTTKAMER, E. **Autonomous Land Vehicles: Steps towards Service Robots**. Deutsche Nationalbibliothek, 2009.

[263] BROGÅRDH, T. **Robot control overview: An industrial perspective**. *Modeling, Identification and Control 30*, 3 (2009), 167–180.

[264] BUEHLER, M., IAGNEMMA, K., AND SINGH, S. **The DARPA urban challenge: autonomous vehicles in city traffic**, vol. 56. Springer Science & Business Media, 2009.

[265] BURGUERA, A., GONZÁLEZ, Y., AND OLIVER, G. **Sonar sensor models and their application to mobile robot localization**. *Sensors 9*, 12 (December 2009), 10217 – 10243.

[266] COURBON, J., MEZOUAR, Y., AND MARTINET, P. **Autonomous navigation of vehicles from a visual memory using a generic camera model**. *Intelligent Transport System (ITS) 10* (2009), 392–402.

[267] GIFFORD, C. M., AKERS, E. L., STANSBURY, R. S., AND AGAH, A. **Mobile robots for polar remote sensing**. In *The Path to Autonomous Robots*, G. Sukhatme, Ed. Springer US, 2009, pp. 1–22.

[268] GONZÁLEZ, J., BLANCO, J., GALINDO, C., DE GALISTEO, A. O., FERNÁNDEZ-MADRIGAL, J., MORENO, F., AND MARTÃ■NEZ, J. **Mobile robot localization based on ultra-wideband ranging: A particle filter approach**. *Robotics and Autonomous Systems 57*, 5 (2009), 496 – 507.

[269] HANFORD, S. D., JANRATHITIKARN, O., AND LONG, L. N. **Control of mobile robots using the soar cognitive architecture**. *JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION 6* (February 2009).

[270] HEIJENK, G. J. **Connect & drive: On the use of cooperative adaptive cruise control to increase traffic stability and efficiency**. In *Third ERCIM Workshop on eMobility, Enschede, The Netherlands* (Enschede, May 2009), no. WP 09-03 in CTIT Workshop Proceedings, Centre for Telematics and Information Technology University of Twente, p. 123.

[271] KRÜGER, J., LIEN, T., AND VERL, A. **Cooperation of human and machines in assembly lines**. *{CIRP} Annals - Manufacturing Technology 58*, 2 (2009), 628 – 646.

[272] LEITNER, J. **Multi-robot cooperation in space: A survey**. In *Advanced Technologies for Enhanced Quality of Life, 2009. AT-EQUAL '09.* (July 2009), pp. 144–151.

[273] MONDADA, F., BONANI, M., RAEMY, X., PUGH, J., CIANCI, C., KLAPTOCZ, A., MAGNENAT, S., ZUFFEREY, J.-C., FLOREANO, D., AND MARTINOLI, A. **The e-puck, a robot designed for education in engineering**. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (Portugal, 2009), vol. 1, pp. 59–65.

[274] RAHOK, S., AND KOICHI, O. **Odometry correction with localization based on landmarkless magnetic map for navigation system of indoor mobile robot**. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on* (Feb 2009), pp. 572–577.

[275] WANG, J., AND SHAN, J. **Segmentation of lidar point clouds for building extraction**. In *Proceedings American Society of Photogramm Remote Sensing Annual Conference, Baltimore, MD, USA* (2009), pp. 9–13.

[276] ANTONELLI, G., ARRICHIELLO, F., AND CHIAVERINI, S. **The nsb control: a behavior-based approach for multi-robot systems**. *Paladyn 1*, 1 (2010), 48–56.

[277] AVANZINI, P., THUILOT, B., AND MARTINET, P. **Urban vehicle platoon using monocular vision: Scale factor estimation**. In *11th International Conference on Control, Automation, Robotics and Vision* (December 2010), Singapure, pp. 1803–1808.

[278] BENZERROUK, A., ADOUANE, L., LEQUIEVRE, L., AND MARTINET, P. **Navigation of multi-robot formation in unstructured environment using dynamical virtual structures**. In *International Conference on Intelligent Robots and Systems (IROS)* (2010), pp. 5589–5594.

[279] BISCHOFF, R., KURTH, J., SCHREIBER, G., KOEPPE, R., ALBU-SCHAEFFER, A., BEYER, A., EIBERGER, O., HADDADIN, S., STEMMER, A., GRUNWALD, G., AND HIRZINGER, G. **The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing**. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)* (June 2010), pp. 1–8.

[280] DÁVILA, A., AND NOMBELA, M. **Sartre: Safe road trains for the environment**. In *Conference on Personal Rapid Transit PRT@ LHR* (2010), vol. 3, pp. 2–3.

[281] GHOMMAM, J., MEHRJERDI, H., SAAD, M., AND MNIF, F. **Formation path following control of unicycle-type mobile robots**. *Robotics and Autonomous Systems 58*, 5 (2010), 727 – 736.

[282] KLEIN WOLTERINK, W., KARAGIANNIS, G., AND HEIJENK, G. J. **A dynamic geocast solution to support cooperative adaptive cruise control (cacc) merging**. In *Fourth ERCIM workshop on e-mobility, Lule?, Sweden* (May 2010), pp. 113–119.

[283] LEVINSON, J., AND THRUN, S. **Robust vehicle localization in urban environments using probabilistic maps**. In *IEEE International Conference on Robotics and Automation* (May 2010), Alaska, USA.

[284] LI, H., NASHASHIBI, F., AND TOULMINET, G. **Localization for intelligent vehicle by fusing mono-camera, low-cost gps and map data**. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (2010), IEEE, pp. 1657–1662.

[285] MARKOFF, J. **Google cars drive themselves, in traffic.**, October 9 2010. The New York Times.

[286] MCKINNEY, R., ZAPATA, M., CONRAD, J., MEISWINKEL, T., AND AHUJA, S. **Components of an autonomous all-terrain vehicle**. In *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the* (March 2010), pp. 416–419.

[287] MOORE, S. W., BOHM, H., AND JENSEN, V. **UNDERWATER ROBOTICS: SCIENCE, DESIGN & FABRICATION**. Marine Advanced Technology Education, 2010.

[288] MYUNG, H., LEE, H.-K., CHOI, K., AND BANG, S. **Mobile robot localization with gyroscope and constrained kalman filter**. *International Journal of Control, Automation and Systems 8*, 3 (2010), 667–676.

[289] SEENI, A., SCHÄFER, B., AND HIRZINGER, G. **Robot mobility systems for planetary surface exploration-state-of-the-art and future outlook: A literature survey**. INTECH Open Access Publisher, 2010.

[290] TAKAHASHI, M., SUZUKI, T., SHITAMOTO, H., MORIGUCHI, T., AND YOSHIDA, K. **Developing a mobile robot for transport applications in the hospital domain**. *Robotics and Autonomous Systems 58*, 7 (2010), 889 – 899. Advances in Autonomous Robots for Service and Entertainment.

[291] VAZ, D. A., INOUE, R. S., AND GRASSI JR., V. **Kinodynamic motion planning of a skid-steering mobile robot using rrts**. In *Proceedings of the 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting* (2010), LARS '10, IEEE Computer Society, pp. 73–78.

[292] ABBADI, A., MATOUSEK, R., AND PETR MINAR, P. S. **Rrts review and options**. In *International Conference on Energy, Environment, Economics, Devices, Systems, Communications, Computers* (2011).

[293] ADOUANE, L., BENZERROUK, A., AND MARTINET, P. **Mobile robot navigation in cluttered environment using reactive elliptic trajectories**. In *18th IFAC World Congress* (August 2011).

[294] BENZERROUK, A. **Architecture de Contrôle Hybride pour les Systèmes Multi-robots Mobiles**. PhD thesis, Ecole Doctorale Sciences pour l'Ingenieur de Clermont-Ferrand, 2011.

[295] CORKE, P. **Robotics, Vision and Control: Fundamental Algorithms in MATLAB**. Springer, 2011.

[296] DASGUPTA, P., WHIPPLE, T., AND CHENG, K. **Effects of multi-robot team formations on distributed area coverage**. *International Journal of Swarm Intelligent Research 2*, 1 (Jan. 2011), 44–69.

[297] ERBS, F., BARTH, A., AND FRANKE, U. **Moving vehicle detection by optimal segmentation of the dynamic stixel world**. In *Intelligent Vehicles Symposium (IV), 2011 IEEE* (June 2011), pp. 951–956.

[298] GRUMERT, E. **Cooperative systems: an overview**.

[299] JIMÉNEZ, F., AND NARANJO, J. E. **Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system**. *Transportation Research Part C: Emerging Technologies 19*, 4 (2011), 658 – 672.

[300] KALLEM, V., KOMOROSKI, A., AND KUMAR, V. **Sequential composition for navigating a nonholonomic cart in the presence of obstacles**. *IEEE Transactions on Robotics 27*, 6 (December 2011), 1152 – 1159.

[301] KARAMAN, S., AND FRAZZOLI, E. **Sampling-based algorithms for optimal motion planning**. *International Journal of Robotics Research 30*, 7 (June 2011), 846–894.

[302] KLANČAR, G., MATKO, D., AND BLAŽIČ, S. **A control strategy for platoons of differential drive wheeled mobile robot**. *Robotics and Autonomous Systems 59*, 2 (2011), 57 – 64.

[303] KLEIN WOLTERINK, W., HEIJENK, G., AND KARAGIANNIS, G. **Automated merging in a cooperative adaptive cruise control (cacc) system**. In *Fifth ERCIM Workshop on eMobility* (Barcelona, Spain, June 2011), Technical University of Catalonia, pp. 23–26.

[304] LATEGAHN, H., GEIGER, A., AND KITT, B. **Visual slam for autonomous ground vehicles**. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (May 2011), pp. 1732–1737.

[305] MICHAEL, N., FINK, J., AND KUMAR, V. **Cooperative manipulation and transportation with aerial robots**. *Autonomous Robots 30*, 1 (2011), 73–86.

[306] MOORE, M. M., AND LU, B. **Autonomous vehicles for personal transport: A technology assessment**. *Available at SSRN 1865047* (2011).

[307] (NASA), N., November 2011. Mars Science Laboratory mission.

[308] PLOEG, J., SERRARENS, A., AND HEIJENK, G. **Connect & drive: design and evaluation of cooperative adaptive cruise control for congestion reduction**. *Journal of Modern Transportation 19*, 3 (2011), 207–213.

[309] SAFEPLATOON. **http://web.utbm.fr/safeplatoon/**, March 2011. Projet ANR VTT SafePlatoon.

[310] SEZEN, B. **Modeling automated guided vehicle systems in material handling**. *Otomatiklestirilmi Rehberli Arac Sistemlerinin Transport Tekniginde Modellemesi, Dou Universitesi Dergisi, 4*, 2 (Feb. 2011), 207 – 216.

[311] SHAMES, I., DEGHAT, M., AND ANDERSON, B. **Safe formation control with obstacle avoidance**. In *IFAC World Congress* (Milan - Italy, 2011).

[312] SIQUEIRA, A. A., TERRA, M. H., AND BERGERMAN, M. **Robust control of cooperative manipulators**. In *Robust Control of Robots*. Springer, 2011, pp. 197–225.

[313] THRUN, S. **GoogleŠs driverless car**. *Ted Talk, Ed* (2011).

[314] TSUGAWA, S., KATO, S., AND AOKI, K. **An automated truck platoon for energy saving**. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (Sept 2011), pp. 4109–4114.

[315] VILCA, J., TERRA, M., AND GRASSI, V. J. **Formation control with leadership alternation for obstacle avoidance.** In *18th World Congress of International Federation of Automatic Control* (Italy, August 2011).

[316] BENZERROUK, A., ADOUANE, L., AND MARTINET, P. **Altruistic distributed target allocation for stable navigation in formation of multi-robot system**. In *10th International IFAC Symposium on Robot Control (SYROCO'12)* (Dubrovnik - Croatia, 5-7, September 2012).

[317] BERGENHEM, C., SHLADOVER, S., COELINGH, E., ENGLUND, C., AND TSUGAWA, S. **Overview of platooning systems**. In *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)* (2012).

[318] BLAZIC, S. **Four-state trajectory-tracking control law for wheeled mobile robots**. In *10th International IFAC Symposium on Robot Control* (September 2012), Croatia.

[319] BONFÈ, M., SECCHI, C., AND SCIONI, E. **Online trajectory generation for mobile robots with kinodynamic constraints and embedded control systems**. In *10th International IFAC Symposium on Robot Control* (September 2012), Croatia.

[320] CHAN, E., GILHEAD, P., JELÍNEK, P., KREJCI, P., AND ROBINSON, T. **Cooperative control of sartre automated platoon vehicles**. In *19th ITS World Congress, Vienna, Austria* (2012), pp. 22–26.

[321] CHAO, Z., ZHOU, S.-L., MING, L., AND ZHANG, W.-G. **UAV formation flight based on nonlinear model predictive control**. *Mathematical Problems in Engineering 2012* (2012), 1–15.

[322] DORIGO, M., FLOREANO, D., GAMBARDELLA, L., MONDADA, F., NOLFI, S., BAABOURA, T., BIRATTARI, M., BONANI, M., BRAMBILLA, M., BRUTSCHY, A., BURNIER, D., CAMPO, A., CHRISTENSEN, A., DECUGNIÈRE, A., DI CARO, G., DUCATELLE, F., FERRANTE, E., FÖRSTER, A., GUZZI, J., LONGCHAMP, V., MAGNENAT, S., GONZALEZ, J. M., MATHEWS, N., DE OCA, M. M., O'GRADY, R., PINCIROLI, C., PINI, G., RÉTORNAZ, P., ROBERTS, J., SPERATI, V., STIRLING, T., STRANIERI, A., STUETZLE, T., TRIANNI, V., TUCI, E., TURGUT, A., AND VAUSSARD, F. **Swarmanoid: a novel concept for the study of heterogeneous robotic swarms**. *IEEE Robotics & Automation Magazine* (2012), in press.

[323] EL-ZAHER, M., CONTET, J.-M., GECHTER, F., AND KOUKAM, A. **Echelon platoon organisation: a distributed approach based on 2-spring virtual links**. In *15th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)* (Germany, 2012).

[324] GU, T., AND DOLAN, J. M. **On-road motion planning for autonomous vehicles**. In *Intelligent Robotics and Applications*, C.-Y. Su, S. Rakheja, and H. Liu, Eds., vol. 7508. Springer Berlin Heidelberg, 2012.

[325] GUPTE, S., MOHANDAS, P., AND CONRAD, J. **A survey of quadrotor unmanned aerial vehicles**. In *Southeastcon, 2012 Proceedings of IEEE* (March 2012), pp. 1–6.

[326] INOUE, R. S. **Controle robusto descentralizado de movimentos coordenados de robôs heterogêneos**. PhD thesis, Universidade de São Paulo, 2012.

[327] LEE, J.-W., AND LITKOUHI, B. **A unified framework of the automated lane centering/changing control for motion smoothness adaptation**. In *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (2012), pp. 282–287.

[328] LINDSEY, Q., MELLINGER, D., AND KUMAR, V. **Construction with quadrotor teams**. *Autonomous Robots 33*, 3 (2012), 323–336.

[329] LOZENGUEZ, G., ADOUANE, L., BEYNIER, A., MOUADDIB, A., AND MARTINET, P. **Map partitioning to approximate an exploration strategy in mobile robotics**. *MAGS - Multiagent and Grid Systems* (2012).

[330] LUETTEL, T., HIMMELSBACH, M., AND WUENSCHE, H.-J. **Autonomous ground vehiclesŮconcepts and a path to the future**. *Proceedings of the IEEE 100*, Special Centennial Issue (May 2012), 1831–1839.

[331] MARTINS, M. M., SANTOS, C. P., FRIZERA-NETO, A., AND CERES, R. **Assistive mobility devices focusing on smart walkers: Classification and review**. *Robotics and Autonomous Systems 60*, 4 (2012), 548 – 562.

[332] MASOUD, A. A. **A harmonic potential approach for simultaneous planning and control of a generic uav platform**. *Journal of Intelligent and Robotic Systems 65*, 1 (2012), 153 – 173.

[333] MURPHY, R. **A decade of rescue robots**. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (Oct 2012), pp. 5448–5449.

[334] PESTEREV, A. V. **Stabilizing control for a wheeled robot following a curvilinear path**. In *10th International IFAC Symposium on Robot Control* (September 2012), Croatia.

[335] RESENDE, P., NASHASHIBI, F., CHARLOT, F., HOLGUIN, C., BOURAOUI, L., AND PARENT, M. **A cooperative personal automated transport system - a citymobil demonstration in rocquencourt**. In *ICARCV 2012 : 12th International Conference on Control, Automation, Robotics and Vision* (Guangzhou, China, 2012), Nanyang Technological University of Singapore, Chinese University of Hong Kong and IEEE Guangzhou Control Chapter, IEEE.

[336] RITZ, R., MÜLLER, M., HEHN, M., AND D'ANDREA, R. **Cooperative quadrocopter ball throwing and catching**. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (Oct 2012), pp. 4972–4978.

[337] RUCCO, A., NOTARSTEFANO, G., AND HAUSER, J. **Computing minimum lap-time trajectories for a single-track car with load transfer**. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on* (2012), pp. 6321–6326.

[338] SCHNEIDER, F., WILDERMUTH, D., AND WOLF, H. **Professional ground robotic competitions from an educational perspective: A consideration using the example of the european land robot trial (elrob)**. In *Intelligent Systems (IS), 2012 6th IEEE International Conference* (Sept 2012), pp. 399–405.

[339] SHARMA, S., AND TAYLOR, M. E. **Autonomous waypoint selection for navigation and path planning: A navigation framework for multiple planning algorithms**. Tech. rep., Arial University, 2012.

[340] ULUSOY, A., SMITH, S., DING, X. C., AND BELTA, C. **Robust multi-robot optimal path planning with temporal logic constraints**. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (May 2012), pp. 4693–4698.

[341] URDIALES, C. **Collaborative Assistive Robot for Mobility Enhancement (CARMEN)**. Springer-Verlag, 2012.

[342] VAN DIJKE, J. P., AND VAN SCHIJNDEL, M. **Citymobil, advanced transport for the urban environment**. *Transportation Research Record: Journal of the Transportation Research Board 2324*, 1 (2012), 29–36.

[343] VOLKSWAGEN, A., AND BMW, V. **Autonomous vehicles**. *Auto Tech Review 1*, 5 (2012).

[344] WANG, R., AND WANG, J. **Fault-tolerant control for electric ground vehicles with independently-actuated in-wheel motors**. *Journal of Dynamic Systems, Measurement, and Control 134*, 2 (2012), 021014.

[345] ZE-SU, C., JIE, Z., AND JIAN, C. **Formation control and obstacle avoidance for multiple robots subject to wheel-slip**. *International Journal of Advanced Robotic Systems 9* (2012), 1–15.

[346] ZHOU, X., AND BI, S. **A survey of bio-inspired compliant legged robot designs**. *Bioinspiration & biomimetics 7*, 4 (2012), 041001.

[347] ADOUANE, L., AUFRÈRE, R., AVANZINI, P., BRESSON, G., CHAPUIS, R., DERUTIN, J.-P., ROYER, E., THUILOT, B., VILCA, J., DAFFLON, B., AND OTHERS. **Safeplatoon: Sûreté de convois de véhicules autonomes**. *Journées Nationales des Communications dans les Transport (JNCT)* (2013).

[348] ALFRAHEED, M., AND AL-ZAGHAMEEM, A. **Exploration and cooperation robotics on the moon**. *Journal of Signal and Information Processing 4* (2013), 253–258.

[349] AVANZINI, P., ROYER, E., THUILOT, B., AND DÉRUTIN, J.-P. **Using monocular visual SLAM to manually convoy a fleet of automatic urban vehicles**. In *IEEE International Conference on Robotics and Automation (ICRA)* (Karlsruhe (Germany), May 2013).

[350] BENZERROUK, A., ADOUANE, L., AND MARTINET, P. **Obstacle avoidance controller generating attainable set-points for the navigation of multi-robot system**. In *IEEE Intelligent Vehicles Symposium* (2013).

[351] BORJA, R., DE LA PINTA, J., ÁLVAREZ, A., AND MAESTRE, J. **Integration of service robots in the smart home by means of upnp: A surveillance robot case study**. *Robotics and Autonomous Systems 61*, 2 (2013), 153 – 160.

[352] BURNS, L. D. **Sustainable mobility: a vision of our transport future**. *Nature 497*, 7448 (2013), 181–182.

[353] CHAN, R. P. M., STOL, K. A., AND HALKYARD, C. R. **Review of modelling and control of two-wheeled robots**. *Annual Reviews in Control 37*, 1 (2013), 89 – 103.

[354] FAGNANT, D., AND KOCKELMAN, K. **Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendation**. In *Eno Foundation.* (2013).

[355] FERNÁNDEZ-MADRIGAL, J.-A., AND CLARACO, J. L. B. **Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods**. 2013.

[356] INOUE, R. S., CERRI, J. P., TERRA, M. H., AND SIQUEIRA, A. A. **Robust recursive control of a skid-steering mobile robot**. In *Advanced Robotics (ICAR), 2013 16th International Conference on* (2013), IEEE, pp. 1–6.

[357] IPDS. **http://ipds.univ-bpclermont.fr**, March 2013. The Institut Pascal Data Sets.

[358] LITMAN, T. **Autonomous Vehicle Implementation Predictions: Implications for Transport Planning**. 2013.

[359] (NHTSA), N. H. T. S. A. **U.s. department of transportation releases policy on automated vehicle development**, May 2013.

[360] SPRINGER, P. J. **Military Robots and Drones: A Reference Handbook**. ABC-CLIO, 2013.

[361] TSUGAWA, S. **An overview on an automated truck platoon within the energy its project**. In *Advances in Automotive Control* (2013), vol. 7, pp. 41–46.

[362] WEI, J., SNIDER, J., KIM, J., DOLAN, J., RAJKUMAR, R., AND LITKOUHI, B. **Towards a viable autonomous driving research platform**. In *Intelligent Vehicles Symposium (IV), 2013 IEEE* (June 2013), pp. 763–770.

[363] ACKERMAN, E. **Korean competition shows weather still a challenge for autonomous cars**, November 2014. IEEE Spectrum.

[364] ALONSO-MORA, J., SIEGWART, R., AND BEARDSLEY, P. **Human-robot swarm interaction for entertainment: from animation display to gesture based control**. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction* (2014), ACM, pp. 98–98.

[365] ARRICHIELLO, F., MARINO, A., AND PIERRI, F. **A decentralized fault tolerant control strategy for multi-robot systems**. In *Proceedings 19th World Congress of the International Federation of Automatic Control* (2014).

[366] BALTES, J., SADEGHNEJAD, S., SEIFERT, D., AND BEHNKE, S. **Robocup humanoid league rule developments 2002–2014 and future perspectives**. *THE 18TH ANNUAL ROBOCUP INTERNATIONAL SYMPOSIUM* (2014).

[367] BARTH, A. **Vehicle Tracking and Motion Estimation Based on Stereo Vision Sequences**. Bayerische Akademie der Wissenschaften, 2014.

[368] BECERRA, H. M., SAGUES, C., MEZOUAR, Y., AND HAYET, J.-B. **Visual navigation of wheeled mobile robots using direct feedback of a geometric constraint**. *Autonomous Robots 37*, 2 (2014), 137–156.

[369] BENZERROUK, A., ADOUANE, L., AND MARTINET, L. **Stable navigation in formation for a multi-robot system based on a constrained virtual structure**. *Robotics and Autonomous Systems 62*, 12 (2014), 1806 – 1815.

[370] BOND, A. H., AND GASSER, L. **Readings in distributed artificial intelligence**. Morgan Kaufmann, 2014.

[371] CABRERA-MORA, F., AND XIAO, J. **Fleet size of multi-robot systems for exploration of structured environments**. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (Sept 2014), pp. 370–375.

[372] CHAN, E. **Sartre automated platooning vehicles**. In *Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to Deployment* (2014).

[373] COGNETTI, M., ORIOLO, G., PELITI, P., ROSA, L., AND STEGAGNO, P. **Cooperative control of a heterogeneous multi-robot system based on relative localization**. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (Sept 2014), pp. 350–356.

[374] DAFFLON, B., AND GECHTER, F. **Making decision with reactive multi-agent systems: A possible alternative to regular decision processes for platoon control issue**. In *Mexican International Conference on Artificial Intelligence* (2014).

[375] ELAKKIYA, A., RAMKUMAR, S., AND EMAYAVARAMBAN, G. **Performance evaluation of mobile sensor network**. *Journal Of Applied Engineering 2*, 8 (2014).

[376] (ESA), E. S. A., November 2014. Rosetta mission.

[377] FAISAL, M., AL-MUTIB, K., HEDJAR, R., MATHKOUR, H., ALSULAIMAN, M., AND MATTAR, E. **Behavior based mobile for mobile robots navigation and obstacle avoidance**. *International Journal of Computers and Applications*, 8 (2014).

[378] FERNANDES, L. C., SOUZA, J. R., PESSIN, G., SHINZATO, P. Y., SALES, D., MENDES, C., PRADO, M., KLASER, R., MAGALHÃES, A. C., HATA, A., PIGATTO, D., BRANCO, K. C., JR., V. G., OSORIO, F. S., AND WOLF, D. F. **Carina intelligent robotic car: Architectural design and applications**. *Journal of Systems Architecture 60*, 4 (2014), 372–392.

[379] GACOVSKI, Z., AND DESKOVSKI, S. **Different control algorithms for a platoon of autonomous vehicles**. *IAES International Journal of Robotics and Automation (IJRA) 3*, 3 (2014), 151–160.

[380] GUILLET, A., LENAIN, R., THUILOT, B., AND MARTINET, P. **Adaptable robot formation control: Adaptive and predictive formation control of autonomous vehicles**. *IEEE Robotics & Automation Magazine 21*, 1 (March 2014), 28 – 39.

[381] GÜTTLER, J., GEORGOULAS, C., LINNER, T., AND BOCK, T. **Towards a future robotic home environment: A survey**. *Gerontology (International Journal of Experimental, Clinical, Behavioural and Technological Gerontology)* (2014).

[382] HICHRI, B., ADOUANE, L., FAUROUX, J.-C., MEZOUAR, Y., AND DOROFTEI, I. **Cooperative lifting and transport by a group of mobile robots**. In *Springer Tracts in Advanced Robotics, from International Symposium on Distributed Autonomous Robotic Systems, DARS 2014* (Korea, 2-5 November 2014).

[383] JAZAR, R. N. **Vehicle Dynamics: Theory and Application**, 2nd ed. 2014.

[384] JESSUP, J., GIVIGI, S., AND BEAULIEU, A. **Robust and efficient multi-robot 3d mapping with octree based occupancy grids**. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on* (Oct 2014), pp. 3996–4001.

[385] KHAN, M. T., AND DE SILVA, C. **Autonomous and market-based fault tolerant algorithms for multi-robot cooperation**. *Journal of Information Science and Engineering 30*, 2 (2014), 483–500.

[386] LAIRD, D., RAPTIS, I., AND PRICE, J. **Design and validation of a centimeter-scale robot collective**. In *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (Oct 2014), pp. 918–923.

[387] LI, H., TSUKADA, M., NASHASHIBI, F., AND PARENT, M. **Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging**. *Intelligent Transportation Systems, IEEE Transactions on 15*, 5 (2014), 2089–2100.

[388] LIU, H., CHENG, L., TAN, M., AND HOU, Z. **Containment control of general linear multiagent systems with multiple dynamic leaders: A fast sliding mode based approach**. *Automatica Sinica, IEEE/CAA Journal of 1*, 2 (April 2014), 134–140.

[389] LU, X.-Y., AND SHLADOVER, S. **Automated truck platoon control and field test**. In *Road Vehicle Automation*, G. Meyer and S. Beiker, Eds., Lecture Notes in Mobility. Springer International Publishing, 2014, pp. 247–261.

[390] MATTOCCIA, S., MACRI, P., PARMIGIANI, G., AND RIZZA, G. **A compact, lightweight and energy efficient system for autonomous navigation based on 3d vision**. In *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on* (Sept 2014), pp. 1–6.

[391] MCLURKIN, J., MCMULLEN, A., ROBBINS, N., HABIBI, G., BECKER, A., CHOU, A., LI, H., JOHN, M., OKEKE, N., RYKOWSKI, J., KIM, S., XIE, W., VAUGHN, T., ZHOU, Y., SHEN, J., CHEN, N., KASEMAN, Q., LANGFORD, L., HUNT, J., BOONE, A., AND KOCH, K. **A robot system design for low-cost multi-robot manipulation**. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (Sept 2014), pp. 912–918.

[392] MEYER, G., AND DEIX, S. **Research and innovation for automated driving in germany and europe**. In *Road Vehicle Automation*, G. Meyer and S. Beiker, Eds., Lecture Notes in Mobility. Springer International Publishing, 2014, pp. 71–81.

[393] MORENO, E., AND CHUNG, S.-Y. **Seadrone: A modular and reconfigurable underwater robot for task optimization**. In *OCEANS 2014 - TAIPEI* (April 2014), pp. 1–7.

[394] OF AUTOMOTIVE ENGINEERS (SAE), S. **Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems**, January 2014.

[395] OLSON, E. **Multi-robot systems and communications limits**. In *IEEE International Conference on Robotics and Automation* (China, May 2014).

[396] PELLEGRINELLI, S., PEDROCCHI, N., MOLINARI-TOSATTI, L., FISCHER, A., AND TOLIO, T. **Multi-robot spot-welding cell design: Problem formalization and proposed architecture**. *Procedia CIRP 21* (2014), 324–329.

[397] PELLENZ, J., JACOFF, A., KIMURA, T., MIHANKHAH, E., SHEH, R., AND SUTHAKORN, J. **Robocup rescue robot league**. *THE 18TH ANNUAL ROBOCUP INTERNATIONAL SYMPOSIUM* (2014).

[398] PETROV, P., AND NASHASHIBI, F. **Modeling and nonlinear adaptive control for autonomous vehicle overtaking**. *Intelligent Transportation Systems, IEEE Transactions on 15*, 4 (2014), 1643–1656.

[399] PONDA, S., JOHNSON, L., GERAMIFARD, A., AND HOW, J. **Cooperative mission planning for multi-uav teams**. In *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds. Springer Netherlands, 2014, pp. 1447–1490.

[400] PRENDI, L., TSENG, S., AND TAM, E. **Sustainability issues for vehicles and fleet vehicles using hybrid and assistive technologies**. In *Environmental Issues in Automotive Industry*, P. Golinska, Ed., EcoProduction. Springer Berlin Heidelberg, 2014, pp. 65–91.

[401] REJAS, J. I., SANCHEZ, A., GARRIDO, J., GLEZ-DE RIVERA, G., AND PEREZ-BORROTO, M. P. **A varying density 3d laser scanner for unmanned ground vehicles mapping and obstacle detection**. In *Design of Circuits and Integrated Circuits (DCIS), 2014 Conference on* (Nov 2014), pp. 1–6.

[402] RODRÍGUEZ, G. S. **Obstacle detection based on laser range data and segmentation for an autonomous vehicle**. Master's thesis, Blaise Pascal University, 2014.

[403] SCHWENDNER, J., AND KIRCHNER, F. **Space robotics: An overview of challenges, applications and technologies**. *K unstliche Intelligenz (KI) 28*, 2 (2014), 71–76.

[404] SIMONIN, O. **Multi-robot systems: from software needs to human monitoring**. In *MUVS 2014 the international workshop on Multi Unmanned Vehicles Systems* (Compiegne, France, Jun 2014).

[405] SIMPKINS, A., AND SIMPKINS, C. **Rescue robotics [on the shelf]**. *Robotics Automation Magazine, IEEE 21*, 4 (Dec 2014), 108–109.

[406] VAGAŠ, M., BALÁŽ, V., SEMJON, J., AND KOUKOLOVÁ, L. **The chosen aspects of self-reconfigurable robots**. In *Applied Mechanics and Materials* (2014), vol. 613, Trans Tech Publ, pp. 279–285.

[407] VIJAYAN, V. P., AND GOPINATHAN, E. **Improving network coverage and life-time in a cooperative wireless mobile sensor network**. In *Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on* (2014), IEEE, pp. 42–45.

[408] WERFEL, J., PETERSEN, K., AND NAGPAL, R. **Designing collective behavior in a termite-inspired robot construction team**. *Science 343*, 6172 (2014), 754–758.

[409] WU, F., LIU, S., AND MU, W. J. **A tracked robot for complex environment detecting**. In *Applied Mechanics and Materials* (2014), vol. 670, Trans Tech Publ, pp. 1389–1392.

[410] WU, Q., WEI, J., AND LI, X. **Research progress of obstacle detection based on monocular vision**. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on* (Aug 2014), pp. 195–198.

[411] ZHANG, X., AND COCQUEMPOT, V. **Fault tolerant control scheme based on active fault diagnosis for the path tracking control of a 4wd electric vehicle**. In *Intelligent Control (ISIC), 2014 IEEE International Symposium on* (Oct 2014), pp. 2189–2195.

[412] ZHU, Y., BIE, D., IQBAL, S., WANG, X., GAO, Y., AND ZHAO, J. **A simplified approach to realize cellular automata for ubot modular self-reconfigurable robots**. *Journal of Intelligent & Robotic Systems* (2014), 1–18.

[413] 4D-VIRTUALIZ. **Cobaye simulator**, Abril 2015. http://www.4d-virtualiz.com.

[414] ALI, A., GARCIA, G., AND MARTINET, P. **The flatbed platoon towing model for safe and dense platooning on highways**. *Intelligent Transportation Systems Magazine, IEEE 7*, 1 (Spring 2015), 58–68.

[415] ARAÚJO, A., PORTUGAL, D., COUCEIRO, M. S., AND ROCHA, R. P. **Integrating arduino-based educational mobile robots in ros**. *Journal of Intelligent & Robotic Systems 77*, 2 (2015), 281–298.

[416] EFFIDENCE. **Effibox**, Abril 2015. http://effistore.effidence.com.

[417] GEONEA, I., UNGUREANU, A., DUMITRU, N., AND RACILĂ, L. **Dynamic modelling of a four legged robot**. In *New Trends in Mechanism and Machine Science*, P. Flores and F. Viadero, Eds., vol. 24 of *Mechanisms and Machine Science*. Springer International Publishing, 2015, pp. 147–155.

[418] HUANG, Y., CHIBA, R., ARAI, T., UEYAMA, T., AND OTA, J. **Robust multi-robot coordination in pick-and-place tasks based on part-dispatching rules**. *Robotics and Autonomous Systems 64* (February 2015), 70–83.

[419] CALIFORNIA PARTNERS FOR ADVANCED TRANSIT AND HIGHWAYS, 2015. http://www.path.berkeley.edu.

[420] MEMON, W. A., AND BILAL, M. **Fully autonomous flammable gases (methanegas) sensing and surveillance robot**. In *International Conference on Artificial Intelligence, Energy and Manufacturing Engineering* (Dubai, January 2015).

[421] PETKOVIĆ, D., SHAMSHIRBAND, S., ANUAR, N., SABRI, A., ABDUL RAHMAN, Z., AND PAVLOVIĆ, N. **Input displacement neuro-fuzzy control and object recognition by compliant multi-fingered passively adaptive robotic gripper**. *Journal of Intelligent & Robotic Systems* (2015), 1–11.

[422] VINE, S. L., ZOLFAGHARI, A., AND POLAK, J. **Autonomous cars: The tension between occupant experience and intersection capacity**. *Transportation Research Part C: Emerging Technologies 52*, 0 (2015), 1 – 14.

[423] YASUDA, G. **Distributed coordination architecture for cooperative task planning and execution of intelligent multi-robot systems**. *Handbook of Research on Advanced Intelligent Control Engineering and Automation* (2015), 407–426.

[424] CHALLENGES, D. **http://www.darpa.mil/about/history/archives.aspx**.

[425] DICTIONARIES, O. **Robotics**. Oxford University Press.

[426] ISLAM, F., NASIR, J., MALIK, U., AYAZ, Y., AND HASAN, O. **Rrt*-smart: Rapid convergence implementation of rrt* towards optimal solution**. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pp. 1651–1656.

[427] THE EUROPEAN LAND-ROBOT TRIAL (ELROB). **http://www.elrob.org**.

Abstract:

Beyond the interest of robotics laboratories for the development of dedicated strategies for single vehicle navigation, several laboratories around the world are more and more involved in the general challenging field of cooperative multi-robot navigation. In this context, this work deals with the navigation in formation of a group of Unmanned Ground Vehicles (UGVs) dedicated to structured environments. The complexity of this Multi-Robot System (MRS) does not permit the direct use of neither classical perception nor control techniques. To overcome this problem, this work proposes to break up the overall control dedicated to the achievement of the complex task into a group of accurate and reliable elementary behaviors/controllers (e.g., obstacles avoidance, trajectory tracking, target reaching, navigation in formation, formation reconfiguration, etc.). These behaviors are linked to different information given by the sensors to the actions of vehicles. To guarantee the performances criteria (e.g., stability, convergence, state errors) aimed by the control architecture, the potentialities of hybrid controllers (which controlling continuous systems in the presence of discrete events) are considered. This control architecture is validated for a single vehicle to perform safe and flexible autonomous navigation using an appropriate strategy of navigation through suitable set of waypoints. This flexible navigation allows different vehicle maneuvers between waypoints (e.g., target reaching or obstacle avoidance) without using any trajectory planning nor replanning. The designed control law based on Lyapunov synthesis guarantees the convergence to assigned waypoint while performing safe trajectories. Furthermore, an algorithm to select suitable waypoints' positions, named Optimal Multi-criteria Waypoint Selection (OMWS), in structured environments while taking into account the safe and reliable vehicle movements, and vehicle constraints and uncertainties is proposed. Subsequently, the control architecture is extended to Multi-Robot Formation (MRF) using a combination of Leader-Follower and behavior-based approaches. An important cooperative MRS issues in this thesis is the dynamic reconfiguration of the formation according to the context of navigation (e.g., to pass from a triangle configuration towards a line if the width of the navigation way is not sufficient). The proposed Strategy for Formation Reconfiguration (SFR) guarantees the stability and the safety of the MRS at the time of the transitions between configuration (e.g., line towards square, triangle towards line, etc.). Therefore, a safe, reactive and dynamic MRF is obtained. Moreover, the degrees of multi-robot safety, stability and reliability of the system are quantified via suitable metrics. Simulations and experiments using urban vehicles (VIPALABs) of the Institut Pascal laboratory allow to perform exhaustive experiments of the proposed control architecture for the navigation in formation of a group of UGVs.

Keywords: Cooperative robotics, Navigation in formation, Dynamic reconfiguration, Control architectures, Hybrid systems, Lyapunov synthesis, Obstacle avoidance.

Résumé :

Plusieurs laboratoires de robotique à travers le monde travaillent sur le développement de stratégies innovantes pour la navigation autonome de véhicules élémentaires ou en convoi. Dans ce contexte, nos travaux de thèse s'inscrivent principalement dans le cadre de la navigation en formation d'un groupe de véhicules dans des environnements structurés. La complexité de ces systèmes multi-robots ne permet pas l'utilisation directe de techniques classiques de perception et/ou de contrôle/commande. Nos travaux ont consisté à décomposer le contrôle global, dédié à la réalisation de la tâche complexe, en un ensemble de comportements/contrôleurs élémentaires précis et fiables (e.g., évitement d'obstacles, suivi de trajectoire, attraction vers une cible, navigation en formation, etc.). Ces comportements lient les différentes informations fournies par les capteurs aux actions des véhicules. Pour garantir les critères de performances imposés à notre architecture de contrôle/commande (e.g., stabilité, robustesse et/ou borner les erreurs maximales), les potentialités des systèmes hybrides ont été considérées. Cette architecture de contrôle a été validée, dans un premier temps, sur des véhicules pris individuellement, en utilisant notamment une stratégie de navigation sûre et flexible utilisant des points de passage. Cette navigation permet au véhicule d'effectuer différentes manœuvres entre ces points de passage (pour éviter par exemple des obstacles dans l'environnement) et ce sans avoir à planifier/re-planifier des trajectoires globales dans l'environnement. Une loi de commande spécifique, permettant une attraction stable (au sens de Lyapunov) et précise vers des cibles statiques ou dynamiques a été par ailleurs développée. Cette loi de commande garantit la convergence du véhicule vers chaque point de passage tout en garantissant des trajectoires sûres. Par ailleurs, un algorithme nommé OMWS (pour *Optimal Multi-criteria Waypoint Selection*) a été proposé pour sélectionner les configurations optimales des points de passage dans l'environnement. Cet algorithme permet de garantir des mouvements sûrs et fiables du véhicules en tenant compte des contraintes et incertitudes liées à la navigation du véhicule. Par la suite, l'architecture de contrôle/commande proposée a été étendue aux systèmes multi-robots en utilisant la combinaison d'une approche leader-suiveur et comportementale. Un important aspect de la navigation multi-robots est la reconfiguration dynamique de la formation en fonction du contexte de la navigation (e.g., passer d'une configuration triangle vers ligne si la largeur de la voie de navigation ne suffisait pas). Ainsi, des stratégies de reconfiguration dynamique ont été proposées, permettant de garantir la sureté de la formation même au moment des transitions entre configurations. Il est à noter par ailleurs que des métriques spécifiques ont été proposées pour quantifier la fiabilité et la robustesse des stratégies multi-robots proposées. Plusieurs simulations et expérimentations avec des véhicules urbains (VIPALABs) nous ont permis de confirmer la viabilité et efficacité des architectures de contrôle/commande proposées pour la navigation en formation d'un groupe de VIPALABs.

Mots-clés : Robotique coopérative, Navigation en formation, Reconfiguration dynamique de la formation, Architectures de contrôle/commande, Systèmes hybrides, Stabilité au sens de Lyapunov, Évitement d'obstacles.