HAL
open science

Functional architecture for automated vehicles trajectory planning in complex environments

David González Bautista

To cite this version:

David González Bautista. Functional architecture for automated vehicles trajectory planning in complex environments. Automatic. Université Paris sciences et lettres, 2017. English. NNT : 2017PSLEM002 . tel-01568505

HAL Id: tel-01568505

https://pastel.hal.science/tel-01568505v1

Submitted on 25 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

ARCHITECTURE FONCTIONNELLE POUR LA PLANIFICATION DES TRAJECTOIRES DES VEHICULES AUTOMATISES DANS DES ENVIRONNEMENTS COMPLEXES

FUNCTIONAL ARCHITECTURE FOR AUTOMATED VEHICLES TRAJECTORY PLANNING IN COMPLEX ENVIRONMENTS

Ecole doctorale n°432

Science des Métiers de l'Ingénieur

Spécialité Informatique temps réel, robotique et automatique

COMPOSITION DU JURY :

M. Jean-Christophe POPIEUL
Université de Valenciennes, Rapporteur

M. José Eugenio NARANJO
Universidad Politécnica de Madrid, Rapporteur

M. Saïd MAMMAR
Université d'Evry, Président

M. Serge BOVERIE
Continental, Examinateur

M. Fawzi NASHASHIBI
INRIA, Examinateur

M. Joshue PEREZ
Tecnalia, Examinateur

M. Vicente MILANES
Renault, Examinateur

Soutenue par:
David **GONZALEZ BAUTISTA**
Le 3 avril, 2017

Dirigée par :
Fawzi NASHASHIBI

To Alida and Julian,
The ones who shaped the person I am today.

To Morgane,
My main pillar in this difficult but gratifying journey.

To Stephanie,
Always beside me, even in the distance.

Contents

1 Introduction **1**
1.1 Motivation . . . 2
1.2 Objectives . . . 3
1.3 Structure of the document . . . 4
1.4 Contributions . . . 5
1.5 Results and publications . . . 6

2 State of the art **11**
2.1 Advanced Driver Assistance Systems . . . 13
2.1.1 Cruise control systems . . . 13
2.1.2 Warning and pre-crash systems . . . 13
2.1.3 Park assistant . . . 14
2.1.4 Night vision systems . . . 14
2.1.5 Lane departure warning / Lane keeping assistance system (LDW/LKAS) . 14
2.1.6 Map supported systems . . . 15
2.1.7 Vehicle interior observation . . . 15
2.1.8 Level 3 automation . . . 16
2.2 Review of Motion Planning Techniques for Automated Vehicles . . . 16
2.2.1 Graph search based planners . . . 18
2.2.1a Dijkstra algorithm . . . 18
2.2.1b A-star algorithm (A*) . . . 18
2.2.1c State Lattice algorithm . . . 19
2.2.2 Sampling based planners . . . 20
2.2.2a Rapidly-exploring Random Tree (RRT) . . . 20
2.2.3 Interpolating Curve Planners . . . 20
2.2.3a Lines and circles . . . 21
2.2.3b Clothoid Curves . . . 21
2.2.3c Polynomial Curves . . . 21
2.2.3d Bézier Curves . . . 22
2.2.3e Spline Curves . . . 23
2.2.4 Numerical Optimization . . . 23
2.2.4a Function optimization . . . 23
2.3 Motion planning developments by ITS groups worldwide . . . 28
2.4 Discussion . . . 32

3 Trajectory planning in static environments **37**
3.1 Problem description . . . 38
3.2 RITS Architecture . . . 39
3.3 Global Planning . . . 40
3.3.1 Global waypoints route planner . . . 41
3.4 Local planning . . . 43
3.4.1 Bezier curves as a basis for trajectory planning . . . 44
3.4.2 Singular points handling . . . 46
3.4.2a Adjacent consecutive intersections . . . 47
3.4.3 Speed planning and trajectory generation . . . 50
3.4.3a Jerk limitation speed planner . . . 51
3.4.3b Quintic bézier speed planner . . . 53
3.4.3c Speed planners comparison . . . 59
3.5 Roundabouts . . . 61
3.5.1 Roundabouts definition . . . 62
3.5.2 Roundabouts constraints . . . 63
3.5.3 Automated Roundabout Algorithm . . . 63
3.6 Conclusions . . . 69

4 Trajectory planning in dynamic environments **73**
4.1 Problem description . . . 74
4.1.1 Proposed approach . . . 76
4.2 Local Planning in dynamic environments . . . 77
4.2.1 Real-time Trajectory Modification . . . 78
4.2.1a Detecting possible collision states . . . 78
4.2.1b Parametric curves for obstacle avoidance . . . 82
4.2.2 Obstacle avoidance/overtaking . . . 85
4.2.2a Avoidance . . . 85
4.2.2b Passing . . . 86
4.2.2c Return to the original path . . . 87
4.2.2d Considerations on overtaking maneuvers . . . 88
4.3 Conclusion . . . 90

5 Validation tests and applications **95**
5.1 Validation tests—simulations . . . 95
5.1.1 Vehicle model . . . 96
5.1.2 Trajectory planning . . . 97
5.1.2a Path planning in straight and curved segments . . . 97
5.1.2b Path planning in roundabouts . . . 98
5.1.2c Speed profile generation . . . 101
5.1.3 Avoidance maneuvers . . . 103
5.2 Validation tests—real platforms . . . 106
5.2.1 The Cyberbus . . . 106
5.2.2 Trajectory planning . . . 107
5.2.2a Path planning in straight and curved segments . . . 108
5.2.2b Path planning in roundabouts . . . 109
5.2.3 Avoidance maneuvers . . . 109
5.3 Shared Control and arbitration applications . . . 114

5.3.1 Literature review 116
5.3.1a Haptic systems 116
5.3.1b Driver state assessment 117
5.3.1c Legal and liability aspect 118
5.3.2 Proposed approach 118
5.3.3 Risk assessment 120
5.3.3a Driving boundaries 121
5.3.3b Fuzzy inference system 123
5.3.4 Shared control 124
5.3.5 Validation tests 126
5.4 Conclusion 128

6 Conclusion **131**
6.1 Contributions to the state of the art 132
6.1.1 Static environments 132
6.1.2 Dynamic environments 132
6.1.3 Shared Control 132
6.2 Future research direction 133

Bibliography **135**

Chapter 1

Introduction

The latest developments in the Intelligent Transportation Systems (ITS) field allow emerging technologies to show promising results at increasing passengers comfort and safety, while decreasing energy consumption, emissions and travel time. This is specially true in the road transportation area thanks to the appearance of automated vehicles, which are significantly aiding drivers by reducing some driving-associated tedious tasks.

Systems like emergency braking with assistance of active suspension [Lin and Ting, 2007], automatic parking [hua Hsu et al., 2008] or blind angle vehicle detection [OMalley et al., 2011] are contributing toward a safer driving. Such systems are known as advanced driver assistance system (ADAS). Fully automated driving capabilities, that is, vehicles able to drive by themselves without human intervention, can be considered as the long-term goal of ADAS systems, being able to integrate all of their functionalities.

Despite of great efforts, fully automated driving still remains unsolved, where the legal framework (currently under development in USA, Europe and Japan), the social acceptance barrier and the technological milestones are some of the challenges yet to be addressed [Anderson et al., 2014]. From the technological perspective, urban environments and the ability to properly guide the autonomous vehicle when negotiating complex situations (i.e. roundabouts or intersections) is one of the important research areas in the field. With these premises, the goal of this Ph.D. work is to design a novel motion planning architecture to deal with current limitations of existing systems. It will rely on several key aspects as real-time trajectory generation, traffic rules, passengers' comfort, among others, being able to handle static and dynamic obstacles that might be in conflict with the trajectory. The functional architecture will be later validated on different scenarios as urban complex environments (i.e. turns, roundabouts and overtaking maneuvers), different platforms (such as simulations, cybercars and electric vehicles) and also in shared control applications.

This thesis has been developed inside the Robotics and Intelligent Transportation Systems (RITS) team/project at the French national institute for research in computer science and control (INRIA, from french: *Institut National de Recherche en Informatique et en Automatique*). The present Ph.D. work has been involved in some European and French initiatives. The FP7 City-Mobil2 project: Undertaking research in the technical, financial and cultural aspects of automated vehicles technology deployment in European cities, the Artemis ECSEL-JU DESERVE project: designs a methodology for ADAS rapid-prototyping and the French ARN CoCoVeA project: designs a cooperative architecture between the driver and the embedded system. In the following, the author explains the motivation, objectives and main contributions of the presented work.

1.1 Motivation

Driving-associated tasks can benefit from the deployment of intelligent systems (i.e. ADAS), making our daily life easier. Efforts in the road transportation field are focused towards safer and more comfortable transport, pushing more and more the development of ADAS. As a matter of fact, recent demonstrations on automated driving are facing complex real scenarios. Implementations from Waymo (Google's automated vehicles spin-off, with more than 2 million hours of automated driving in different environments—from 2009 to 2016[1]), Daimler and Karlsruhe Institute of Technology (KIT), recreating the first cross-country automobile journey in history, in an autonomous way [Ziegler et al., 2014b]; and Vislab and the University of Parma, showing automated driving capabilities in the PROUD project demo on 2013 [Broggi et al., 2013a], are some of the recent demonstrations with promising results regarding automated driving technologies.

Despite of some remarkable results obtained up to now, there is still a long way to go before making the transition between automated vehicles (i.e. vehicles with some autonomous features) and autonomous vehicles on public roads (i.e. fully autonomous driving). The social acceptance and confidence on the technology, the technical long-term developments yet to address and legal/liability unsolved challenges represent the most important milestones to cover nowadays in the automated vehicles field [Van Schijndel-de Nooij et al., 2011].

The development of intelligent planning algorithms is one of the key issues when designing robotic architectures [Hwang and Ahuja, 1992], [Elbanhawi and Simic, 2014]. The associated constrains in the automated vehicle field—i.e., traffic rules, high speeds or road layout—turn real-time motion planning in a critical aspect for achieving automated driving on complex environments, remaining still as one of the important topics in ITS research [LaValle, 2006], [Hardy and Campbell, 2013]. Motion planning has been extensively investigated in the literature, where the first implementations in automated vehicles date back to the 90's decade in the Eureka PROMETHEUS project [Behringer and Muller, 1998] and the ARGO project [Broggi et al., 1999], where interpolation with clothoids and splines, respectively, were implemented as the main planning techniques.

Many efforts have focused in path planning with obstacle avoidance capabilities for automated vehicles as 2004-2005 Darpa Grand Challenge and 2007 Darpa Urban Challenge (see Carnegie Mellon University—CMU—vehicle [Ferguson et al., 2008], Stanford vehicles [Thrun et al., 2006], [Montemerlo et al., 2008], AnnieWAY team vehicle [Ziegler et al., 2008], among others), paving the road for future oncoming automated vehicle research. The European Research Council (ERC) also demonstrated interest on such initiatives by funding the OFAV project (Open intelligent systems for Future Autonomous Vehicles) demonstrating intercontinental automated vehicle capabilities in the Vislab International Autonomous Challenge (VIAC) [Bertozzi et al., 2011] from Parma to Shanghai. Also the Hyunday-Kia Autonomous Challenge (2010 and 2012) created the scenario for important approaches from Asian teams [Jo et al., 2013], [Ryu et al., 2013]. The main difficulty in the design of these architectures is found when dealing with dynamic obstacles, augmenting the dimensionality of the problem, thus decreasing the algorithm's ability to re-plan its strategy in real time [Hardy and Campbell, 2013]. In this Ph.D. work, the author classifies planning algorithms in four main groups: Graph-based, sampling-based, interpolation and numerical optimization.

Despite the increasing maturity of motion planning algorithms, passenger's comfort-oriented planning architectures have not been further investigated. Not only comfort represents one of the key steps to future social acceptance of these technologies, but also its improvement is one of the goals in the ITS field [Villagra et al., 2012]. The work in [Labakhua et al., 2008] presented a framework to reduce lateral and longitudinal accelerations according to ISO 2631-1 [ISO, 1997].

[1] https://www.waymo.com/ontheroad/

The implementation included a curve interpolation planner, implementing clothoids at its core, producing continuous curvature paths and a speed planning algorithm which sets speed profiles to comply with longitudinal accelerations (limited acceleration ramps) and lateral accelerations (maximum speed set according to the maximum curve's curvature). Villagra et.al. [Villagra et al., 2012] also based their approach on lateral comfort accelerations from [ISO, 1997], implementing a jerk and acceleration bounded speed planner over a clothoid based interpolating path planner. It allowed to have constrained lateral and longitudinal accelerations once the limit speed is found for each curve and then compare the results with human drivers. However, the planner was off-line and could not handle obstacles.

Social acceptance of these motion planning techniques is increased when comfort is part of the design constraints of the systems. However technological gaps will force a transition period from mere ADAS to fully automated driving. One of the ways to make this transition is to share the control with the driver [Van Schijndel-de Nooij et al., 2011]. This keeps the driver (and his/her driving skills) in the control loop while automation vigilance can be added, increasing comfort (reduce of the workload) and security (warning and emergency actions performed by the embedded system). However, it remains unclear how this shared control system will effectively exchange the control with the driver. Some efforts aim to develop systems where the control sharing is "on-off" by determining stability envelopes as in [Erlien et al., 2016]. Others aim to share control with different levels of authority and control for each of the decision makers, smoothly shifting authority in the control action with respect to lane keeping applications [Saleh et al., 2013] and also introducing human machine cooperation examples or metaphors as in [Flemisch et al., 2014] and [Da Lio et al., 2015]. Another related issue is how the action suggestion is going to be provided [Li et al., 2012]. To address this matter, recent research points in the direction of applying the aid "as needed" rather than "all the time" [Flemisch et al., 2008a], [Petermeijer et al., 2015], thus avoiding drivers' over-reliance on automation. Planning algorithms can come to aid this kind of systems by defining the limit regions of where warnings or further active actions should be taken according to collision avoidance [Hardy and Campbell, 2013] and human behavior [Bosetti et al., 2015].

Having this in mind, the present Ph.D. thesis proposes a novel functional architecture for automated vehicles trajectory planning. It will be capable to cope with complex scenarios (e.g. turns, roundabouts and overtaking maneuvers) and provide minimal curvature paths. Its real-time trajectory generation capabilities will allow to avoid obstacles (static and dynamic) in conflict with the original strategy. Suitable speed profiles will be generated by a novel speed planning algorithm, once the geometrical path has been set, allowing the constraint of global accelerations to preserve passengers' comfort. This architecture will be tested in simulated and real scenarios (such as simulations, cybercars and other electric vehicles) to validate the approach, as well as in shared control applications in simulated environments, keeping the driver in the control loop and showing the modularity of the system.

1.2 Objectives

The main objective of this Ph.D. work is the design of a generic modular architecture for automated vehicles trajectory planning . It will implement and improve curve interpolation techniques in the motion planning literature by including comfort as the main design parameter. It will be able to generate—in real-time—suitable geometric trajectories that consider measurements incertitude from the perception system, vehicle's physical limits, the road layout and traffic rules.

Starting from previous developments in RITS team/project at INRIA [Bouraoui et al., 2011], [González and Perez, 2013], the modular architecture is in charge of defining physical geometric

trajectories that can handle complex scenarios as turns, roundabouts and overtaking maneuvers. Comfort and real-time planning capabilities are the two main concepts implemented in the design of the architecture. Curve interpolation techniques are used, firstly to define the geometrical path to be followed by the vehicle according to physical limitations and road layout, minimizing the curvature profile at all moments. Then, a novel speed planner algorithm is implemented to comply with comfort limits. It will limit the longitudinal acceleration perceived by the passengers when planning changes in the speed profile and also limit the maximum speed in each curve to avoid lateral accelerations, thus increasing comfort.

Once this first trajectory is set, an obstacle avoidance module is responsible to account for the information coming from the perception stage and change, in real-time, the trajectory if the original one is conflicted with any obstacle (previously done off-line in [Petrov and Nashashibi, 2013]). This will be done in three main phases: Lane change, passing and returning to the original trajectory. Static and dynamic obstacles will be considered as well as comfort constraints to generate the new set of trajectories that overtake the obstacle. The validation of the proposed functional architecture will be done in simulated and real scenarios.

In order to show the modularity of the presented architecture, validation experiments will also be tested in simulated scenarios with control sharing applications. However, to ultimately share the control with the driver, the computation of the navigable area is necessary. Once the optimal trajectory is found, an obstacle free corridor will be defined taking into account obstacles in the path and comfort limits (as in ISO 2631-1 Standard [ISO, 1997]). It will allow the automated vehicle to assess the performance of the driver with respect to the prediction of future vehicle states and the time to exit this comfort and safe driving region. This envelope will serve as the first step to establish a communication bridge between the decision makers.

With the proposed system, the author aims to improve real-time planning strategies to tackle current state of the art system's limitations. Comfort, traffic rules, road layout, obstacles and vehicle's physical limits are considered at all times by the real-time trajectory planning system. Moreover, a driving envelope provided by the system permits the interaction with the driver in control sharing ADAS, thus enabling the autonomous system to gradually integrate into the urban landscape, cooperating with humans in the driving task.

1.3 Structure of the document

The present Ph.D. work is organized in 6 chapters. These will describe in detail the two stages conforming the cascade system, as well as a thorough state of the art in motion planning techniques for automated vehicles. The remaining chapters are as follows:

Chapter 2, State of the Art: This chapter presents a review of automated vehicles literature, divided in two main blocks. First, the revision of current ADAS in the market is presented. It provides an overview of the tendencies and current technological developments towards more intelligent vehicles. Then, a review of motion planning techniques is given. The different existing techniques implemented in the intelligent ground vehicles literature are presented, as well as a description of each technique used by research teams, their contributions in motion planning and a comparison among them. Relevant works in the overtaking and obstacle avoidance maneuvers are presented, allowing the understanding of open challenges and future milestones.

Chapter 3, Trajectory planning in static environments: Presents the proposed functional motion planning architecture for automated vehicles. It is based on two main modules:

The global and local planners. The global planner sets singular points which describe the characteristics of the road layout, i.e. road stretches, intersections and roundabouts. The local planner takes these singular points and smooths the path between them, implementing curve interpolation techniques to achieve minimal curvature paths with G^1 continuity. A novel speed planning algorithm generates G^2 continuous speed profiles, taking into account the curvature profile of the physical path. Roundabouts, considered as special intersections, are addressed with a three step planning process (i.e. entry, circulatory roadway and exit). The global and local planners are modular and have been adapted to INRIA base architecture for automated vehicle navigation.

Chapter 4, Trajectory planning in dynamic environments: Introduces an obstacle avoidance component in the local planning module of the functional architecture. It is capable of supervising the original trajectory and create collision free trajectories in real-time, when future collisions are detected. Obstacles are considered to be static or dynamic. The avoidance maneuver implemented is the overtaking, divided in three main steps: avoidance, passing and return to the original path. The generated trajectories are G^2 continuous and have minimal curvature profiles, increasing comfort in the avoidance maneuver. The proposed component is modular, intervening only in the communication buffer with the control stage.

Chapter 5, Validation tests and applications: Presents the validation tests and applications for the proposed functional motion planning architecture. Validation test are presented in simulated scenarios (Pro-sivic) and in a real platform of Rits team (the cyber-bus). Results from Chapter 3 are presented in complex scenarios comprising intersections and roundabouts. Obstacle avoidance maneuvers with respect to static and dynamic obstacles are also validated. Moreover, the proposed functional architecture is tested and validated in control sharing applications. The aim is to keep the driver in the control loop, evaluate the risk in the current situation and send haptic signals in case the risk increases. Risk evaluation will be based on the time to cross the limits of the drivable area (determined by the proposed plannign architecture) and the time to collision with other obstacles. Haptic feedback is the common language between decision makers and increases proportionally to the risk in the situation in hand, guiding the driver to the optimal trajectory determined by the embedded system.

Chapter 6, Conclusions and future research direction: Conclusions and most important remarks, with respect to the problems addressed in the present Ph.D. work, are given in this chapter. Also future research lines are presented and discussed.

1.4 Contributions

The main contributions in the present dissertation are as follows:

1. The main contribution of this Ph.D. work is a generic and modular architecture for automated vehicles trajectory planning. The presented architecture improves real-time planning strategies, focusing on current state of the art system's limitations. It will be capable to provide a minimum curvature path to be followed by the vehicle, considering comfort limits, traffic rules, road layout, obstacles and ego-vehicle's physical limits. A novel speed planer is also proposed, capable to provide a smooth speed profile and account for lateral and longitudinal acceleration constraints to increase passengers' comfort. This will allow to have a smooth trajectory to be followed by the automated vehicle.

2. Once a first trajectory is available, the obstacle avoidance component will take into account the information coming from the perception stage and determine if there are future collision states. In case a future collision is detected, the component is able to change—in real-time—the current trajectory and avoid the obstacle in front. The avoidance action to implement is the overtaking maneuver, divided in three steps (avoidance, passing and return to the original path). It will permit to avoid obstacles in conflict with the current trajectory of the ego-vehicle, considering comfort limits and developing a new trajectory that keeps lateral accelerations at its minimum.

3. The modular architecture will be tested in simulated and real environments, validating the approach. Since modularity will also be tested by implementing the general architecture in a shared control ADAS approach, the navigable area is determined by geometrical paths (based in comfort limits) coming from the functional motion planning architecture. This envelope will describe not only the limits of the road, but comfort limits in real-time. With this in mind, a control sharing ADAS is presented, allowing the evaluation of drivers' performance to smoothly share the control through haptic signals. The implementation of the proposed planning architecture in control sharing applications, avoids liability/legal issues since the driver always stays in control, tackles social acceptance by allowing users to get comfortable with technology and not over-rely on it and proposes a way of transition between nowadays ADAS and automated vehicles deployment in the near future.

1.5 Results and publications

The present Ph.D. thesis has been developed within the framework of two main European initiatives: FP7 CityMobil2 project and Artemis ECSEL-JU DESERVE project. These have allowed the collaboration between different research groups and universities worldwide, as well as live demonstrations in different European locations. As results from the work in the development of the thesis, the author cites the most important ones:

Journal articles

Title: A Review of Motion Planning Techniques for Automated Vehicles
Authors: D. González, J. Pérez, V. Milanés, and F. Nashashibi
Journal: Transactions on Intelligent Transportation Systems, IEEE
Volume: 17, **Pages:** 1135-1145, **Year:** 2016

Title: Parametric-Based Path Generation for Automated Vehicles at Roundabouts
Authors: D. González, J. Pérez and V. Milanés
Journal: Expert Systems with Applications
Status: Accepted—in press.

Conference papers

Title: Development and Design of a Platform for Arbitration and Sharing Control Applications
Authors: J. Pérez, D. González, F. Nashashibi, G. Dunand, F. Tango, N. Pallaro and A. Rolfsmeier

Proceedings: International IEEE Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)
Place: Samos, Greece **Date:** July, 2014

Title: Continuous Curvature Planning with Obstacle Avoidance Capabilities in Urban Scenarios
Authors: D. González, J. Pérez, R. Lattarulo, V. Milanés and F. Nashashibi
Proceedings: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)
Place: Qingdao, China **Date:** October, 2014

Title: Low-Speed Cooperative Car-Following Fuzzy Controller for Cybernetic Transport Systems
Authors: V. Milanés, M. Marouf, J. Pérez, D. González and F. Nashashibi
Proceedings: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)
Place: Qingdao, China **Date:** October, 2014

Title: Description and Technical specifications of Cybernetic Transportation Systems: an urban transportation concept
Authors: R. Luis, J. Pérez, D. González and V. Milanés
Proceedings: International IEEE Conference on Vehicular Electronics and Safety (ICVES)
Place: Yokohama, Japan **Date:** November, 2015

Title: Optimized trajectory planning for Cybernetic Transportation Systems
Authors: F. Garrido, D. González, V. Milanés, J. Pérez and F. Nashashibi
Proceedings: 9th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)
Place: Leipzig, Germany **Date:** July 2016

Title: Real-time Planning for Adjacent Consecutive Intersections
Authors: F. Garrido, D. González, V. Milanés, J. Pérez and F. Nashashibi
Proceedings: 19th International IEEE Conference on Intelligent Transportation Systems (ITSC)
Place: Rio de Janeiro, Brazil **Date:** November 2016

Title: Speed Profile Generation based on Quintic Bézier Curves for Enhanced Passenger Comfort
Authors: D. González, V. Milanés, J. Pérez and F. Nashashibi
Proceedings: 19th International IEEE Conference on Intelligent Transportation Systems (ITSC)
Place: Rio de Janeiro, Brazil **Date:** November 2016

Book Chapters

Title: Vehicle Control in ADAS Applications: State of the Art
Authors: J. Pérez, D. González and V. Milanés
Book: John Wiley & Sons, Intelligent Transport Systems: Technologies and Applications
Year: 2015

Title: Arbitration and Sharing control strategies in the driving process
Authors: D. González, J. Pérez, V. Milanés, F. Nashashibi, M. Saez and A. Cuevas
Book: Artemis ECSEL-JU DESERVE project Book
Year: 2016—In press—

Chapitre

Etat de l'art

Below is a French summary of the following chapter "State of the art".

Le développement de la recherche sur les véhicules intelligents permettra, à terme, d'augmenter la sécurité sur les routes. Une partie des développements concerne les systèmes avancés d'aide à la conduite (ADAS) qui aident les conducteurs en réduisant les erreurs humaines [Anderson et al., 2014]. Aujourd'hui, les développements ADAS ont fortement pénétré le marché automobile tout public, offrant aux particuliers des outils pour une conduite plus sûre et confortable. Cependant, les systèmes ADAS ne proposent pas le calcul d'une trajectoire permettant de décrire le chemin que le véhicule va parcourir en temps réel. Ceci est primordial si on veut atteindre le niveau 5 du standard SAE-J3016. La création de la trajectoire nécessite la prise en compte d'un ensemble de contraintes, comme les règles de circulation, la configuration de la route, la vitesse maximale, etc. Ce chapitre passe en revue :

- les caractéristiques des systèmes ADAS disponibles actuellement.
- Les différentes approches permettant d'effectuer une planification de trajectoire en présentant d'abord les principaux algorithmes et en décrivant les techniques implémentées et testées par les principaux groupes de recherche mondiaux travaillant sur les véhicules autonomes.

Chapter 2

State of the art

Intelligent vehicles are reaching maturity and, with them, road security is increasing. A traffic safety report from the National Highway Traffic Safety Administration (NHTSA) shows that from 2003 to 2012 fatalities in the U.S. decreased 27% and 14% in rural and urban areas respectively [NHTSA, 2014]. Europe shows the same trend, having a significant reduction in fatalities of about 18% since 2010[1]. This is directly linked to current developments in Advanced Driver Assistance Systems (ADAS), which are aiding drivers by reducing human errors [Anderson et al., 2014].

Recent initiatives launched between governments and manufactures are currently creating the framework for the new standards and regulations for automated driving. As a matter of fact, some ongoing European initiatives address legal matters and promote the standardization of these technologies, as Citymobil2 project [van Dijke and van Schijndel, 2012]. They got permission from local authorities for vehicle demonstrations, and some certification procedures were developed. In United States, already eight states[2] (Nevada, Florida, California, Michigan, Louisiana, North Dakota, Tennessee, Utah) and the District of Columbia recognize the importance of these technologies for future sustainable mobility and have enacted specific laws to allow automated driving in common traffic [Walker Smith, 2013]. Moreover the U.S. department of transportation (DoT) in a joint effort with the national highway traffic safety administration (NHTSA) issued in September 2016 a federal automated vehicles policy [Federal Automated Vehicles Policy, 2016], setting design and development guidelines for automated vehicles as a first step for a future and more complete regulatory framework.

Recently, a new taxonomy of automated driving levels was issued by SAE International [SAE, 2014], where five different automated levels were identified (see Figure 2.1 for details). Other levels of automation have already been proposed by the German Federal Highway Research Institute (BASt) [Gasser and Westhoff, 2012] and the National Highway Traffic Safety Administration [NHTSA, 2013]. A comparison of these is summarized in [SAE, 2014], stating that the SAE taxonomy is alike the other two, but gives a broader and more specified view of higher automation levels (see Figure 2.1). As a matter of fact, the SAE-J3016 standard has been taken into account by automated vehicles development projections and first regulatory works in the U.S. (DoT) and Europe, as the leading definition for automation levels (see [Federal Automated Vehicles Policy, 2016] and [ERTRAC, 2015]). Figure 2.2 presents examples of U.S. and European research tracks, issued in 2015 (Figure 2.2b) and 2016 (Figure 2.2a) respectively. These aim to provide a roadmap and a

[1] Road safety in the European Union, online at:
http://ec.europa.eu/transport/road_safety/pdf/vademecum_2015.pdf

[2] Autonomated – Self-Driving vehicles legislation, online at:
http://www.ncsl.org/research/transportation/autonomous-vehicles-legislation.aspx#NCSL Publications

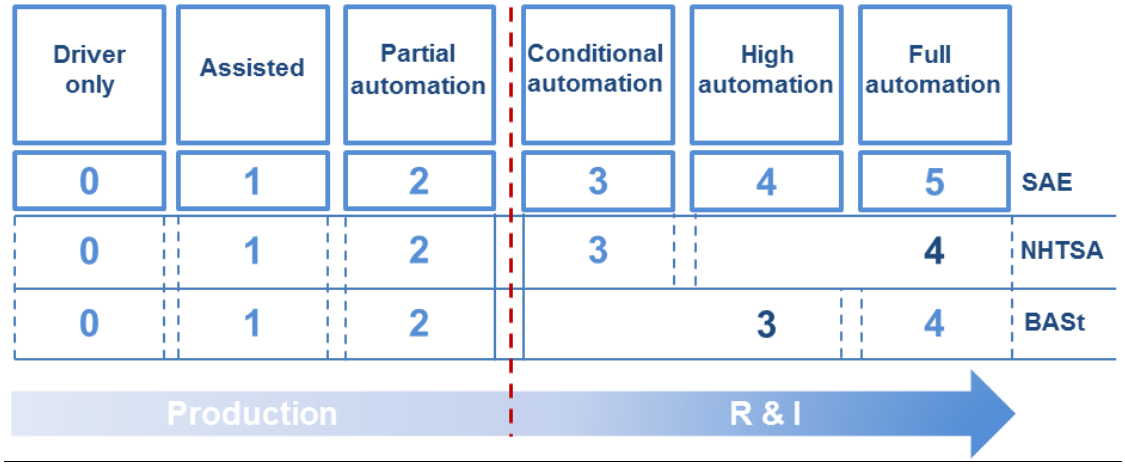
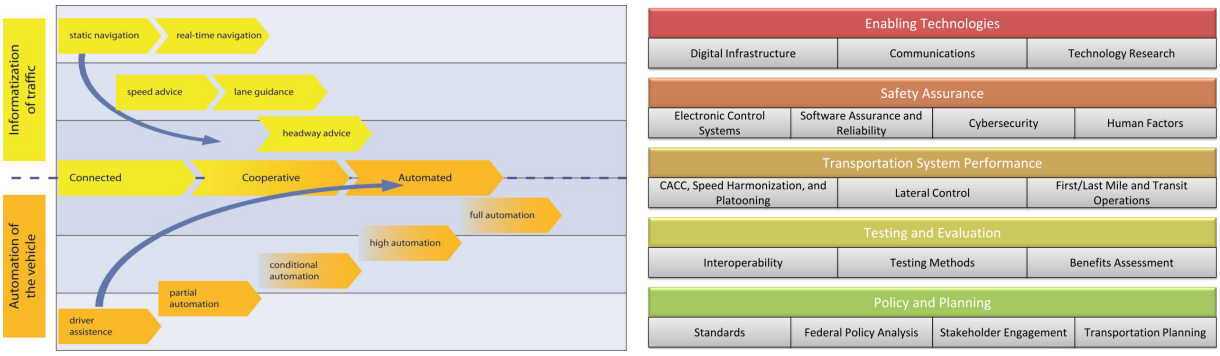


Figure 2.1: Summary of levels of driving automation for on-road vehicles [SAE, 2014], image from [Dokic et al., 2015]



(a) European research track [European Commision, 2016] (b) U.S. DoT research track [Dopart, 2015]

Figure 2.2: European and U.S. main research approaches for automated vehicles development

broad view of future developments for these technologies. Broad reviews of current actions worldwide towards the maturity of intelligent vehicles can be found in [ERTRAC, 2015] and [Trimble et al., 2014].

Automated vehicle industry (ADAS development) has moved from passive ADAS (i.e. warning and information to the driver), to more active ADAS (as LKAS and ACC systems). Today, the trend is to introduce vehicle cooperation (e.g. CACC systems [Milanés et al., 2014b]) and intelligence in the decision making process to raise the automation level "stair", decreasing driver's workload an increasing safety and comfort. However, the decision stage remains still out of on-the-market systems, remaining the driver in the control loop for all on-the-mark ADAS (i.e. levels 1 and 2 from SAE taxonomy)

Having this in mind, the present chapter gives an overview of current state-of-the-art in automated vehicles; it is divided in two blocks. First, a review of the most recent ADAS in the market is carried out, showing the clear direction on the research field toward more sophisticated and intelligent vehicles. Then, an in-deep review of the motion planning research field in automated driving is presented. Important groups worldwide are also reviewed, focusing in the description of

the planning techniques applied in real demonstrations. An overview of the tendencies and orientation of the state-of-the-art according to research centers and industry developments is included, concluding with remaining challenges, current research direction from important researchers and manufactures, including future steps in motion planning developments.

2.1 Advanced Driver Assistance Systems

Some international research groups and vehicle manufacturers are technologically ready to provide fully autonomous driving. However, the complexity of traffic scenarios, some legal constraints and the driver's acceptance, forecast a soft transition between manual and fully autonomous driving. In this context, the automation capacities in Advanced Driver Assistance Systems (ADAS) have a very important role. These can be defined as those active safety systems which require some monitoring on the vehicle's environment and on driver intentions. This extra information is combined with ego-vehicle data (positions and speed profile) in order to provide the driver with some warning or intervene with automatic actuation with the goal of increasing safety. Some of the most important applications currently available in the automotive market or soon be introduced are described below:

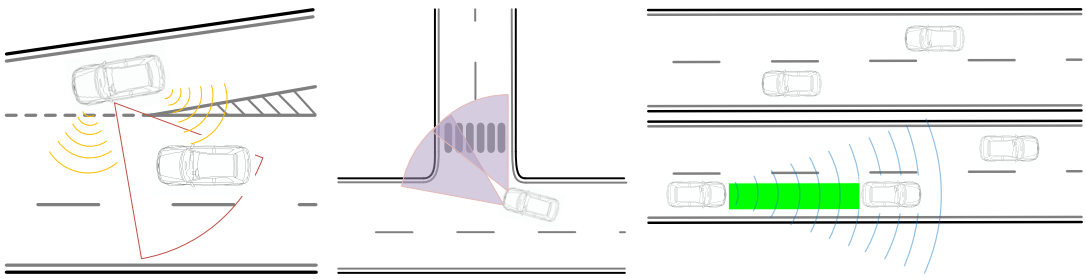
2.1.1 Cruise control systems

The Cruise Control (CC) systems are capable to keep automatically the speed of the vehicle. Firsts CC implementations were based on controlling the acceleration actuator only (longitudinal control). Adaptive Cruise Control (ACC), one of the most conventional forms of ADAS, was the next step, acting over the longitudinal control of the vehicle, permitting it to follow a leader and to maintain a predefined head-time (see Figure 2.3c). The next step in the evolution of this technology is based on cooperation among different vehicles in order to reduce the headway between vehicles and the accordion effect in traffic jams. This is known as Cooperative ACC (CACC), and it is based on Vehicle to Vehicle (V2V) and in some cases Vehicle to Infrastructure (V2I) communications [Milanés et al., 2014b]. The system works usually between 30 and 180 km/h. The maximum deceleration provided by the system is in between 2 and $3m/s2$. ACC Stop & Go improves the performance of the conventional ACC to a full stop capability. The stop and go of the vehicle is, thus, automatically performed, so the range of the system is extended.

2.1.2 Warning and pre-crash systems

These systems are related to forward detection, mainly using radars, lasers, cameras, infrared sensors and, in some cases, fusing different sensors. Although most of them have automatic actions, it is not mandatory. The forward looking system involves three main modules: object detection, decision making and actuation [Llorca et al., 2011]. Warning systems as Frontal Collision Warning (FCW) or Collision Warning Systems (CWS), implement object detection and Human-Machine Interfaces (HMI), remaining the driver as the main responsible to avoid a potential collision. This system is extended by Automated Emergency Braking (AEB) and Collision Mitigation by Braking (CMbB). The former is activated if it is still possible to brake before the collision; the latter when there is an imminent non-avoidable crash. Both, AEB and CMbB actuate only over the brake.

Pre-crash systems can be divided according to their deceleration in: 1) Soft braking, up to 5 m/s2; and 2) Hard braking, from 5 m/s2 to the full capability of the braking system. In some cases is possible to provide a progressive braking: first, a soft braking can be provided and, if no action is taken by the driver, a hard braking is applied. Also, a pre-fill of the brake circuit (when



(a) Blind spot detection (b) Light control when turning (c) Adaptive cruise control

Figure 2.3: Example of some on-the-market ADAS

the FCW is launched) can be provided to be ready for a full-brake action, either by the driver or the embedded system.

2.1.3 Park assistant

It is probably the most used (and demanded) ADAS today. Ultrasound park assist systems have evolved from high-end to mainstream vehicles in few years. These systems can help in the parking maneuver in close-fitting spaces, by alerting the driver of rear obstacles and their distance to the vehicle. The intelligent park assist provides easy parking by identifying sufficient parking spaces and steering the car into it. The system is always supervised by the driver, who can override the operation pushing the accelerator pedal or the brake pedal. Other parking assistant systems use rear view camera in addition to the ultrasound sensors. They provide a video image from the rear area of the vehicle. Lateral and longitudinal controllers are used simultaneously in these systems.

2.1.4 Night vision systems

Night Vision Systems (NVS) permit drivers to see in low or difficult light conditions. When weather conditions are extreme, these systems can see beyond the range illuminated by the headlights of the vehicle. The technology is based on near and far infrared cameras, illuminating the road ahead, along a spectrum invisible to the human eyes, then shown on the board display. Many manufactures are using these technologies (Mercedes-Benz, Toyota, Audi, BMW, among others). Recently, the light vision system came off second-best in preference for car consumer option in Europe [Lee and Yu, 2012]. The night vision system has four basic functions: pedestrian detection, pedestrian collision warning, image display and sound warning. These systems use the information from an image which is composed with the thermal radiation of objects.

Many premium vehicle brands offer different night vision systems. Most recent generation night vision systems have added pedestrian detection as a feature to assist drivers to avoid potential collisions. These are classified in near-infrared (NIR) and far-infrared (FIR) according to the regions of the electromagnetic spectrum [Luo et al., 2010].

2.1.5 Lane departure warning / Lane keeping assistance system (LDW/LKAS)

The Lane Departure Warning system constantly checks if the vehicle is not invading lane markings (without using the blinkers). Many Original Equipment Manufacturers (OEM) offer today lane

departure systems under different commercial brands (AFIL, Audi Lane Assist, etc.). It is composed by a sensor (or several sensors) with the capability to detect when the driver is leaving from the chosen lane, a control unit and a suitable HMI for the driver head-up display[3]. Line detection can be done through two different technologies: Infrared sensors (e.g. placed in the low part of the vehicle—PSA models—) or image processing (e.g. a camera usually placed behind the windshield, on the rear view mirror housing, etc). The system works from a certain speed (commonly between 60 and 80 km/h upwards). Moreover, when activating the suitable blinker, the system understands that the driver really wants to change lane and no warning is provided in case of crossing the lines. Lane Keeping Assistant Systems (LKAS) are an extension to LDW systems, which includes an additional torque on the steering wheel (electrical power steering is required) that helps the driver to keep the vehicle into the desired lane.

As a complementary system for LDW and LKAS, Blind Spot Detection (BSD) systems have the goal to warn the driver in case another vehicle is located in the blind spot which is not visible by the rear-view mirrors (see Figure 2.3a). Therefore, it counts on some sensors (commonly, short range radars with frequencies close to $24GHz$ or image processing units) constantly monitoring the area placed in the lateral blind spots of the vehicle. These sensors provide information to the driver with a warning, which can be acoustic, visual or haptic.

2.1.6 Map supported systems

Architectures of map-supported ADAS are described in [Durekovic and Smith, 2011]. They explain that digital map data can be classified in three levels: non-map ADAS, map-enhanced ADAS and map-enabled ADAS. An example of the first one is the ultrasound parking distance control, whereas those for the second ones are the ACC and Speed Limit Info (SLI). These systems work without digital map information, but their functionalities can be improved with the addition of digital map data. Curve Speed Warning (CSW) and Dynamic Pass Predictor (DPP) [Loewenau et al., 2006] are examples of systems that need digital map inputs. CWS accounts for the curvature of the following stretch and plans a speed limit profile to address the curve. DPP gives information to the driver of whether is permitted or not to overtake in the current road stretch. At night, map support can be used to adapt the headlamps to the scenario (see Figure 2.3b). Adaptive high-beam assist can turn in the sense of the curves, anticipating to possible undesirable obstacles (e.g. a bicycle or pedestrians) or even modify the height of high-beams to avoid blinding other drivers[4].

2.1.7 Vehicle interior observation

Driver drowsiness is one of the major causes of road accident. For this reason, several driver drowsiness detection systems have been implemented to warn the driver in this dangerous situation. Some of them are based on eyelid detection and also gaze detection (direction) [Hu and Zheng, 2009], a non-invasive detection. Other manufactures are using biomedical signals (e.g. FICOSA), which permit to characterize drivers' alertness to the driving task, issuing different alarms if fatigue is detected. In [Sahayadhas et al., 2012], drowsiness detection is classified in: 1) Vehicle based measures (Deviation from lane's center); 2) Behavioral measures (yawning, eye closure, eye blink and head pose); and 3) Physiological measures (e.g. electrocardiogram signals), describing the pros and cons of each method.

Driver distraction is another mayor cause of accidents. As a matter of fact, current approaches can profit from gaze detection to implement FCW or AEB applications as in [Lee et al., 2011].

[3]Example of an LDW HMI: http://continental-head-up-display.com/

[4]http://techcenter.mercedes-benz.com/en/adaptive_high_beam_assist/detail.html

Manufacturers such as Continental have developed visual distraction detection prototypes with inner cameras, to detect inattention or distraction from the driving task (gaze detection) [Boverie et al., 2011].

2.1.8 Level 3 automation

Recently, several OEMs as Tesla, Mercedes, Infinity and BMW focused their efforts in level 3 automation[5]. The idea profits from different ADAS such as ACC, LKAS, AEB, among others, to create a more complete and adaptable system that can handle different situations, reaching conditional automation driving in highways. Such systems—"Autopilot" (Tesla) or "DRIVE pilot" (Mercedes-Benz), are described as beta systems and the driver has to remain attentive at all times to appropriately respond in case a request to intervene is issued. The core of these systems (comprehending different sensors such as radar, cameras, ultrasound sensors, among others) is a lane marking and obstacle detection system able to stay within the lane actuating in the steering wheel and keeping its distance from frontal obstacles in highways. Up to now, only Tesla's system can perform automated lane changes.

Despite of the great amount of ADAS commercially available today, there is still a long way to go before having fully automated vehicles on the road. One of the unsolved challenges is the ability to develop a real-time motion planning algorithms. Trajectory planning algorithms play a key role for moving SAE's taxonomy to the latest levels [SAE, 2014]. Private initiatives as Google's self-driving vehicles, nuTonomy[6] self-driving cab service in Singapore and Uber's own version in Pittsburgh[7] are the first to test complete level 5 systems with decision and motion planning included. However, they remain as pilot programs and ongoing research has much to cover before having robust level 5 vehicles everywhere.

Next section presents a review of motion planning techniques where the most used algorithms are described, including pros-cons analysis. Then, a review of the techniques used by the most significant research groups worldwide in the field is carried out.

2.2 Review of Motion Planning Techniques for Automated Vehicles

Automated vehicles development was very limited before the 90s because reduced investments in the field [Shladover et al., 1991]. Thanks to the evolution of information technology applicable to vehicle automation, the ITS concept was created. Different research centers around the world were oriented to this end (e.g. California PATH, Parma University, among others), improving intelligent vehicle systems. An overview of some important development events in automated navigation systems is presented in Section 2.3.

A description of the first automated vehicles—dating back to the late 80's and beginning of the 90's (see also Figure 2.9)—are mentioned by [Shladover et al., 1991] and [Behringer and Muller, 1998]. [Shladover et al., 1991] describe longitudinal control systems (including vehicle following control, inter-vehicular communications and a comparison between different methods) and lateral control systems (considering vehicle lateral dynamics and magnetic sensors as path reference with no path planning involved), introducing the Advance Highway Systems (AHS) concept. [Behringer and Muller, 1998] describe the architecture proposed for the VaMoRs-L vehicle

[5] Available online at: http://goo.gl/yw75q3
[6] Available online at: http://nutonomy.com/
[7] Available online at: https://newsroom.uber.com/pittsburgh-self-driving-uber/

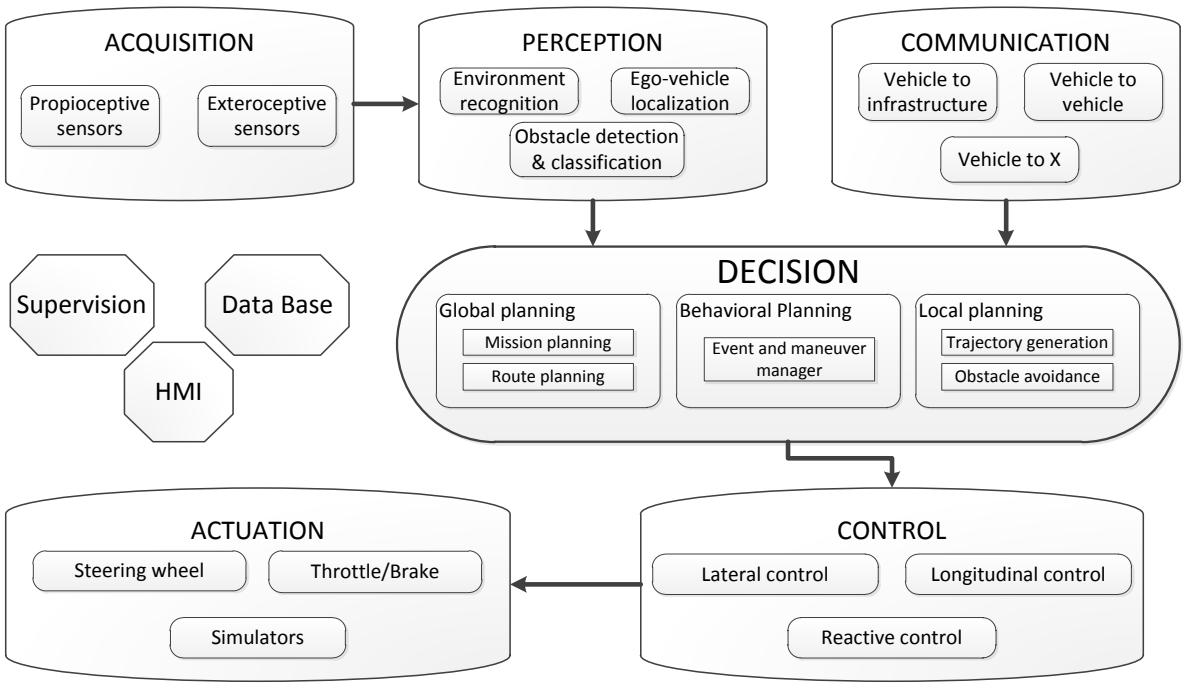


Figure 2.4: Control architecture for automated vehicles. General abstraction based on: [Ziegler et al., 2014b], [Broggi et al., 2013a], [Urmson et al., 2008], [Thrun et al., 2006], [Ferguson et al., 2008], [Jo et al., 2014] and [Jo et al., 2015].

in the PROMETHEUS project. A vehicle that was able to perform automated driving, aided by vision and path generation algorithms.

When it comes to develop automated vehicles, a literature review shows different proposed architectures. Stanford [Thrun et al., 2006] and Carnegie Mellon University (CMU) [Urmson et al., 2008] pioneered the implementation of fully automated vehicles in the Grand and Urban DARPA challenges. In Europe, the VIAC project [Bertozzi et al., 2011] and Daimler with KIT [Ziegler et al., 2014b] described the different processing stages needed for the control architecture of automated vehicles, where perception, decision and control are the most important. Figure 2.4 shows a general architecture for automated vehicles that can be extracted from earlier developments, including interconnections based on previous real automated vehicle implementations. Following these first implementations, different architectures have been proposed for automated driving. Figure 2.4 presents a summary of the main architectures where perception, decision and control are the three main components of the software configuration. The main focus of this Ph.D. work is related to the decision component of the architecture, specifically motion planning techniques in automated vehicles.

Motion planning has been extensively investigated in the literature, starting from the first developments in mobile robotics applications [Han et al., 2008]. It is considered a key aspect for robot navigation since it provides global and local trajectory planning to describe the behavior of the robot. It considers the dynamic and kinematic models of the robot to go from a starting position to a final one. The main difference between vehicles and robots for carrying out motion planning are in the fact that the former addresses road networks where traffic rules have to be obeyed; whereas the latter has to cope with open environments without much specific rules to

follow, only to reach the final destination.

Most authors divide the problem into global and local planning. A review of the different approaches and concept definitions (as global, local or reactive motion planning) can be found in [Kunchev et al., 2006,Hwang and Ahuja, 1992] and [Elbanhawi and Simic, 2014]. A great amount of navigation techniques have been taken from mobile robotics and modified to face the challenges of road networks and driving rules. These planning techniques were classified in four groups, according to their implementation in automated driving: graph search, sampling, interpolating and numerical optimization (see Table 2.1). The most relevant path planning algorithms implemented in motion planning for automated driving are described in the following.

2.2.1 Graph search based planners

In automated driving, the basic idea is to traverse a state space to get from point A to point B. This state space is often represented as an occupancy grid or lattice that depicts where objects are in the environment. From the planning point of view, a path can be set implementing graph searching algorithms that visit the different states in the grid, giving a solution (that not necessarily is the optimal one) or not (there is no possible solution) to the path planning problem. Some of these algorithms have been applied to the automated vehicles development.

2.2.1a Dijkstra algorithm

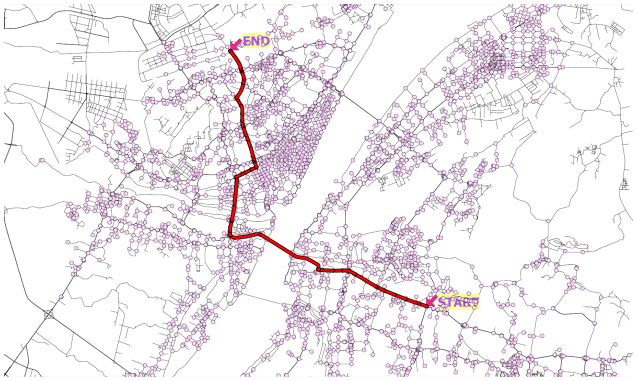
It is a graph searching algorithm that finds single-source shortest path in the graph. The configuration space is approximated as a discrete cell-grid space, lattices, among others (e.g. [Marchese, 2006], [LaValle and Hutchinson, 1998]).

Description of the concept and implementation of the algorithm can be found in [LaValle, 2006] and [Hwang et al., 2003]. It searches the closest non-searched nodes in a grid until the goal is found, taking the least expensive path. In automated driving, it has been implemented by [Kala and Warwick, 2013] in multi-vehicles simulations, in the Little Ben vehicle (Ben Franklin racing team's entry to the Darpa Urban Challenge [Bohren et al., 2008]) and also team VictorTango [Bacha et al., 2008].An urban implementation is shown in Figure 2.5a [Li et al., 2009].

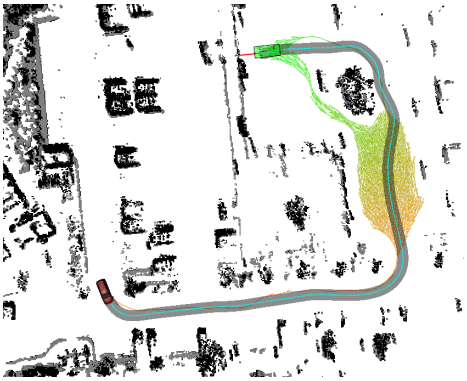
2.2.1b A-star algorithm (A*)

It is a graph searching algorithm that enables a fast node search due to the implementation of heuristics (it is an extension of Dijkstra's graph search algorithm). Its most important design aspect is the determination of the cost function, which defines the weights of the nodes. It is suitable for searching spaces mostly known *a priori* by the vehicle [Likhachev and Ferguson, 2009], but costly in terms of memory and speed for vast areas.

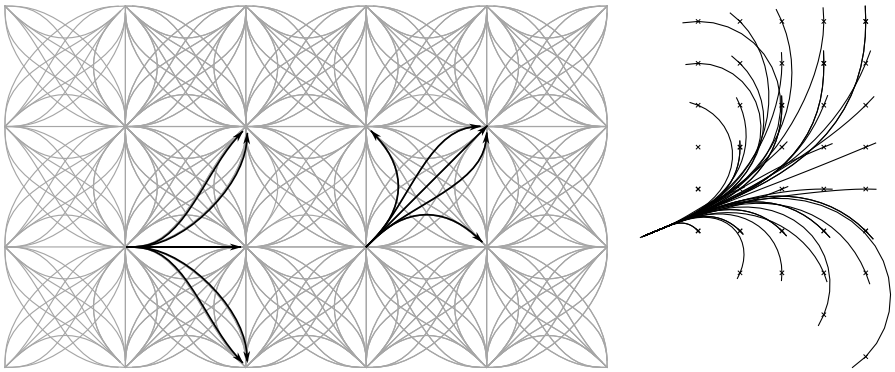
Several applications in mobile robotics have used as basis for improvement, such as the dynamic A* (D*) [Stentz, 1994], Field D* [Ferguson and Stentz, 2006], Theta* [Nash et al., 2007], Anytime repairing A* (ARA*) and Anytime D* (AD*) [Likhachev et al., 2008], among others. Ziegler et.al. [Ziegler et al., 2008] implemented the A* algorithm along with Voronoi cost functions for planning in unstructured spaces and parking lots. In [Montemerlo et al., 2008] (hybrid A*) and [Kammel et al., 2008] (A*), it served as part of the DARPA Urban Challenge in Junior (Stanford University automated vehicle, see Figure 2.5b) and AnnieWAY (KIT) respectively. In [Ferguson et al., 2008] AD* was used by Boss, the winning vehicle of the DARPA Urban Challenge.



(a) Dijkstra [Li et al., 2009]



(b) A* [Montemerlo et al., 2008]



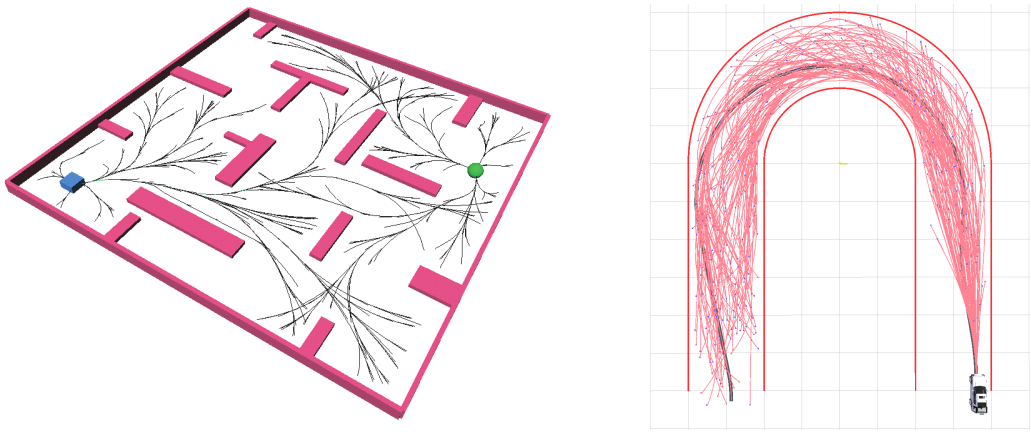
(c) Lattices [Ziegler and Stiller, 2009]

Figure 2.5: Graph-search based planning algorithms as presented in the literature. a) Representation of a global path using the Dijkstra algorithm by [Li et al., 2009], where intersections are represented by 2D nodes, thus enabling the graph search. b) Hybrid A* as implemented in the DARPA Challenge by Junior [Montemerlo et al., 2008]. c) Lattices and motion primitives as presented in [Ziegler and Stiller, 2009].

2.2.1c State Lattice algorithm

The algorithm uses a discrete representation of the planning area with a grid of states (often a hyper-dimensional one). This grid is referred as *state lattice* over of which the motion planning search is applied [Pivtoraiko and Kelly, 2005]. The path search in this algorithm is based in local queries from a set of lattices or primitives containing all feasible features, allowing vehicles to travel from an initial state to several others. A cost function decides the best path between the precomputed lattices. A node search algorithm is applied in different implementations (e.g. A* in [Kushleyev and Likhachev, 2009] or D* in [Rufli et al., 2009]).

Howard and Kelly applied state lattices in rough terrains for a wheeled mobile robot [Howard and Kelly, 2007], where simulations give promising results of global and local planning for spacial missions, as well as in [Pivtoraiko et al., 2009]. Table 2.1 shows the authors reference for automated driving implementations divided in state and spatio-temporal lattices. Most of the works consider time and speed dimensions as [Ziegler and Stiller, 2009] and [McNaughton et al., 2011]. A good example is presented in Figure 2.5c from [Ziegler and Stiller, 2009], where the non-temporal lattice is presented on the left and a finer discretization, on the right, also includes curvature.



(a) RRT with kinematic constraints [LaValle and Kuffner, 2001]

(b) RRT* with kinematic and dynamic constraints [Jeon et al., 2013]

Figure 2.6: Rapidly-exploring Random Trees (RRT). a) RRT including kinematic constraints from [LaValle and Kuffner, 2001]. b) RRT* as showed in [Jeon et al., 2013].

2.2.2 Sampling based planners

These planners try to solve timing constrains—i.e. planning in high dimensional spaces—that deterministic methods can't meet. The approach consists on randomly sampling the configuration space or state space, looking for connectivity inside it [Elbanhawi and Simic, 2014]. The downside is the fact that the solution is suboptimal. The most commonly used methods in robotics are the Probabilistic Roadmap Method (PRM) [Kavraki et al., 1996] and the Rapidly-exploring Random Tree (RRT) [LaValle and Kuffner, 2001]. The latter has been extensively tested for automated vehicles.

2.2.2a Rapidly-exploring Random Tree (RRT)

It belongs to the sampling-based algorithms applicable to on-line path planning [Karaman and Frazzoli, 2011]. It allows a fast planning in semi structured spaces [LaValle and Kuffner, 2001] by executing a random search through the navigation area. It also has the ability to consider non-holonomic constraints (such as maximum turning radius, see Figure 2.6a).

In [Elbanhawi and Simic, 2014] and [Karaman and Frazzoli, 2011] descriptions, applications and improvements of these algorithms are reviewed. In automated driving, it has been used for the MIT team at DARPA Urban Challenge [Kuwata et al., 2009]. However, the resulting path is not optimal most of the time (although the solution is improved through time), jerky and not curvature continuous. In [Karaman and Frazzoli, 2010] a new approach of this algorithm was developed, named RRT* (See Figure 2.6b from [Jeon et al., 2013]). This new implementation converges to an optimal solution but having the same disadvantages. First results are shown in [Karaman et al., 2011]. Further developments and implementations in automated vehicles are mentioned in Table 2.1.

2.2.3 Interpolating Curve Planners

Techniques as Computer Aided Geometric Design (CAGD) are often used as path smoothing solutions for a given set of way-points [Brezak and Petrovic, 2014]. These allow the motion

planners to fit a given description of the road by considering feasibility, comfort, vehicle dynamics and other parameters in order to draw the trajectory.

Interpolation is defined as the process of constructing and inserting a new set of data within the range of a previously known set (reference points). This means that these algorithms take a previously set of knots (e.g. a given set of way-points describing a global road map), generating a new set of data (a smoother path) in benefit of the trajectory continuity, vehicle constraints and the dynamic environment the vehicle navigates [Labakhua et al., 2008]. In the presence of obstacles, it suffices to generate a new path to overcome the event and then re-entry the previously planned path. The interpolating curve planners implement different techniques for path smoothing and curve generation, being the most common in the automated driving field those indicated in Table 2.1 and 2.2 (see also Figure 2.7f-j).

2.2.3a Lines and circles

Different segment road network can be represented by interpolating known way-points with straight and circular shapes. It is a simple mathematical method to approach the planning problem in car-like vehicles [Reeds and Shepp, 1990], [Horst and Barbera, 2006]. An example is presented in Figure 2.7a for the shortest path to turn around a vehicle to perform forward and backward driving.

2.2.3b Clothoid Curves

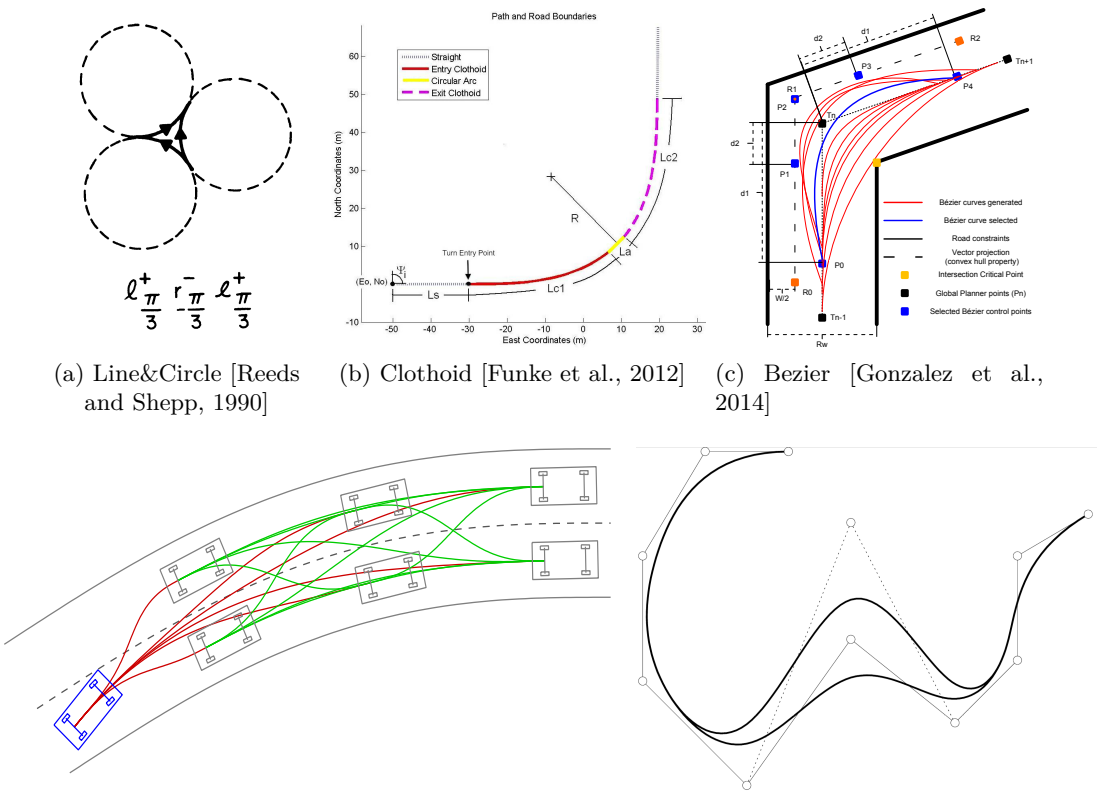
This type of curve is defined in terms of Fresnel integrals [Brezak and Petrovic, 2014]. Using clothoid curves is possible to define trajectories with linear changes in curvature since their curvature is equal to their arc-length; making smooth transitions between straight segments to curved ones and vice versa(see Figure 2.7b [Funke et al., 2012]). Clothoids have been implemented in the design of highways and railways and are also suitable for car-like robots [Walton and Meek, 2005].

The implementation of clothoid primitives was tested in the VIAC project. In [Broggi et al., 2012] the current curvature is taken from the steering wheel position and then the other profiles are evaluated taking into consideration the dynamic limitations (e.g. fishtailing) and the physical ones (e.g. the steering wheel).

2.2.3c Polynomial Curves

These curves are commonly implemented to meet the constraints needed in the points they interpolate, i.e. they are useful in terms of fitting position, angle and curvature constraints, among others. The desired values or constraints in the beginning and ending segment will determine the coefficients of the curve. For the computation of the polynomial coefficient, the reader is referred to [Simon and Becker, 1999], [Piazzi et al., 2002] and [Glaser et al., 2010].

The Laboratory on Interactions between Vehicles, Infrastructure and Drivers (LIVIC) [Glaser et al., 2010] implemented these curves for line change scenarios. Fourth degree polynomials where used for longitudinal constraints and fifth degree polynomials for the lateral constraints, meeting the desired parameter for different scenarios. Also in [Petrov and Nashashibi, 2014] cubic polynomials where used to generate safe trajectories for overtaking maneuvers. In [Xu et al., 2012], cubic (green line) and quartic (red line) polynomial curves are implemented with a state lattice planner (see Figure 2.7d).



(a) Line&Circle [Reeds and Shepp, 1990]

(b) Clothoid [Funke et al., 2012]

(c) Bezier [Gonzalez et al., 2014]

(d) Polynomial [Xu et al., 2012]

(e) Spline [Farouki, 2008]

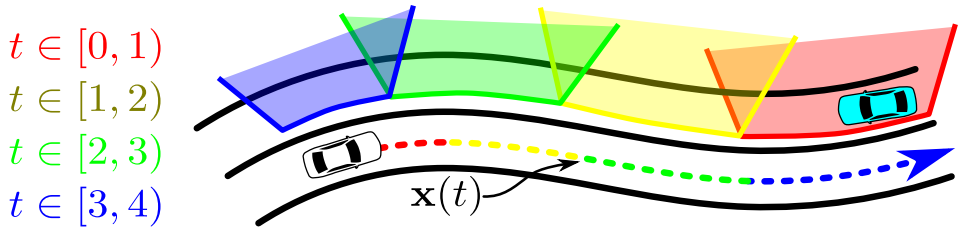
Figure 2.7: Interpolating Curve Planners. a) Optimal path to turn the vehicle around as proven in [Reeds and Shepp, 1990]. b) Planning a turn for the Audi TTS from Stanford [Funke et al., 2012]. c) Evaluation of several Bézier curve in a turn, as showed in [Gonzalez et al., 2014]. d) Different motion states, planned with polynomial curves as presented in [Xu et al., 2012]. e) Spline behavior when a knot changes place, as presented in [Farouki, 2008].

2.2.3d Bézier Curves

These are parametric curves that rely on control points to define their shape. The core of Bézier curves are the Bernstein polynomials. These curves have been extensively used in CAGD applications, technical drawing, aeronautical and automotive design.

The advantage of this kind of curves is their low computational cost, since the curve behavior its defined by control points. The constraints at the beginning and the end of the curvature can be met by correctly placing these control points according to different properties described in [Perez et al., 2014, Choi et al., 2008].

A good example of the modularity and malleability of Bézier curves is presented in [Walton et al., 2003], interconnecting circular shapes and also interconnecting Bézier curves with continuous curvature profiles in [Yang and Sukkarieh, 2010]. These curves are often used to approximate clothoid curves [Wang et al., 2001, Sánchez-Reyes and Chacón, 2003], or implementing rational Bézier curves for fast planning as in [Montes et al., 2008, Montes et al., 2007]. In [Perez et al., 2014, Han et al., 2010] and [Gonzalez et al., 2014] 3rd and 4th degree Béziers (see Figure 2.7c) are implemented in automated vehicles, evaluating the best applicable curve to the situation in hand (i.e. turns, roundabouts, lane change, obstacle avoidance, among others).



$$J[\mathbf{x}] = \int_{t_0}^{t_0+T} j_{\text{offs}} + j_{\text{vel}} + j_{\text{acc}} + j_{\text{jerk}} + j_{\text{yawr}} \, \mathrm{d}t$$

Figure 2.8: Function Optimization (of $J_{[X]}$ in this example) presented by [Ziegler et al., 2014a].

2.2.3e Spline Curves

A spline is a piecewise polynomial parametric curve divided in sub-intervals that can be defined as polynomial curves [Piazzi et al., 2002], [Bacha et al., 2008], b-splines [Shiller and Gwo, 1991], [Berglund et al., 2010] (that can also be represented in Bézier curves [Romani and Sabin, 2004]) or clothoid curves [Walton and Meek, 2005]. The junction between each sub-segment is called knot and they commonly possess a high degree of smoothness constraint at the joint between the pieces of the spline. An example of a b-spline with a changing knot is shown in Figure 2.7e.

2.2.4 Numerical Optimization

These methods aim to minimize or maximize a function subject to different constrained variables. In path planning is often used to smooth previously computed trajectories as in [Dolgov et al., 2010] and also to compute trajectories from kinematic constraints [Ziegler et al., 2014b].

2.2.4a Function optimization

This technique finds a real valued function's roots (minimize the variables outcome). It has been implemented to improve the potential field method (PFM) in mobile robotics for obstacles and narrow passages, obtaining C_1 continuity [Ren et al., 2006]. [Ziegler et al., 2014b] and [Ziegler et al., 2014a] aim to find C_2 continuous trajectories by minimizing a function that considers position, velocity, acceleration and jerk as planning parameters (see Figure 2.8).

In Table 2.1, a taxonomy of motion planning techniques implemented in automated vehicles is presented. It classifies the planning technique in each contribution according to the previously described algorithms. Table 2.2 complements Table 2.1 , giving a comprehensive review about the pros and cons of each planning technique.

These implementations show that the first steps for automated driving have already been set. Different architectures implement motion planners in order to have a safe and continuous path to follow. To mitigate or eliminate risky situations (such as dynamic obstacle avoidance, emergency situations, intersection and merging negotiation, among others) is the current focus in the literature. The consideration of perception limitations (uncertainty in measurements) and control needs for safety and comfort are key aspects in futures steps. Some works have already considered such parameters, by generating driving corridors and trajectory proposals that can be communicated to the driver by an HMI [Glaser et al., 2010], [Hardy and Campbell, 2013]. Others consider the uncertainty in data acquisition [Xu et al., 2014], communications [Lee and Park, 2012] and even future behavior prediction [Havlak and Campbell, 2014] to avoid situations like the MIT-Cornell collision in the Darpa Urban Challenge [Fletcher et al., 2008].

Table 2.1: Taxonomy of motion planning techniques applied in automated driving scenarios

Technique	Technique description	Implemented in
Graph-Search Based Planners:		
Dijkstra's Algorithm	Known nodes/cells search space with associated weights	[Bohren et al., 2008], [Chen et al., 2008], [Patz et al., 2008], [Anderson et al., 2012]
	Grid and node/cells weights computation according to the environment	[Bacha et al., 2008], [Kala and Warwick, 2013]
A* algorithm family	Anytime D* with Voronoi cost functions	[Urmson et al., 2008], [Ferguson et al., 2008], [Likhachev and Ferguson, 2009]
	Hybrid-heuristics A*	[Montemerlo et al., 2008], [Dolgov et al., 2010]
	A* with Voronoi/Lattice environment representation	[Kammel et al., 2008], [Ziegler et al., 2008], [Chen et al., 2008]
	PAO* as in [Ferguson et al., 2004]	[Miller et al., 2008]
State Lattices	Environment decomposed in a local variable grid, depending on the complexity of the maneuver.	[Ferguson et al., 2008], [Likhachev and Ferguson, 2009], [Pivtoraiko and Kelly, 2005], [Howard et al., 2008]
	Spatio-temporal lattices (considering time and velocity dimensions)	[McNaughton et al., 2011], [Gu et al., 2013], [Ziegler and Stiller, 2009], [Werling et al., 2011], [Kushleyev and Likhachev, 2009], [Furgale et al., 2013]
Sampling-Based Planners:		
Rapidly-exploring Random Trees	Physical and logical bias are used to generate the random-tree	[Kuwata et al., 2009], [Braid et al., 2006], [Ryu et al., 2013]
	Anytime planning with RRT*	[Karaman et al., 2011], [Jeon et al., 2013], [Anderson et al., 2012], [Aoude et al., 2010].
	Trajectory coordination with RRT	[Kala and Warwick, 2011]

Table 2.1: Taxonomy of motion planning techniques applied in automated driving scenarios

Technique	Technique description	Implemented in
Interpolating Curve Planners:		
Line and circle	Road fitting and interpolation of known waypoints	[Reeds and Shepp, 1990], [Hsieh and Ozguner, 2008]
Clothoid Curves	Piecewise trajectory generation with straight, clothoid and circular segments	[Fraichard and Scheuer, 2004], [Funke et al., 2012], [Broggi et al., 2012], [Vorobieva et al., 2013], [Vorobieva et al., 2014], [Fuji et al., 2014]
	Off-line generation of clothoid primitives from which the best will be taken in on-line evaluation	[Brezak and Petrovic, 2014], [Coombs et al., 2000], [Behringer and Muller, 1998]
Polynomial Curves	Cubic order polynomial curves	[Petrov and Nashashibi, 2014], [McNaughton et al., 2011]
	Higher order polynomial curves	[Piazzi et al., 2002], [Glaser et al., 2010], [woo Lee and Litkouhi, 2012], [Simon and Becker, 1999], [Keller et al., 2011]
Bézier Curves	Selection of the optimal control points location for the situation in hand	[Perez et al., 2014], [Gonzalez et al., 2014], [Pérez et al., 2013], [Han et al., 2010], [Simon and Becker, 1999], [Liang et al., 2012]
	Rational Bézier curves implementation	[Montes et al., 2008], [Montes et al., 2007]
Spline Curves	Polynomial piecewise implementation	[Thrun et al., 2006], [Piazzi et al., 2002], [Bacha et al., 2008]
	Basis splines (b-splines)	[Trepagnier et al., 2006], [Shiller and Gwo, 1991], [Berglund et al., 2010]

Table 2.1: Taxonomy of motion planning techniques applied in automated driving scenarios

Technique	Technique description	Implemented in
Numerical Optimization Approaches:		
Function optimization	Trajectory generation optimizing parameters such as speed, steering speed, rollover constraints, lateral accelerations, jerk (lateral comfort optimization), among others	[Cremean et al., 2006], [Kogan and Murray, 2006], [Dolgov et al., 2010], [Ziegler et al., 2014b], [Ziegler et al., 2014a], [Gu and Dolan, 2012]

Table 2.2: Comparison of benefits and disadvantages in motion planning techniques

Technique	Advantages	Disadvantages
Dijkstra's algorithm	Finds the shortest path in a series of nodes or grid. Suitable for global planning in structured and unstructured environments (Figure 2.5a).	The algorithm is slow in vast areas due to the important amount of nodes. The search is not heuristic. The resulting path is not continuous. Not suitable for real time applications.
A* family	Based on the Dijkstra algorithm. The search is heuristic reducing computation time (Figure 2.5b).	The resulting path is not continuous. The heuristic rule is not straightforward to find most of the times.
State lattices	Able to handle several dimensions (position, velocity, acceleration, time). Suitable for local planning and dynamic environments (Figure 2.5c).	Computationally costly due to the evaluation of every possible solution in the database. The planner is only resolution complete (lattice discretization).
RRT familly	Able to provide a fast solution in multi-dimensional systems. The algorithm is complete and always converges to a solution (if there is one and given enough time). Suitable for global and local planning, see Figure 2.6.	The resulting trajectory is not continuous and therefore jerky. The optimality of the path strongly depends of the time frame for the RRT* case.
Interpolating curve planner	Optimization of the curvature and smoothness of the path is achieved through the implementation of CAGD techniques (compared here below). Suitable for local planning oriented to comfort and safety in structured environments.	Depends on a global planning or global waypoints. Time consuming when managing obstacles in real time because the optimization of the path and consideration of road an ego-vehicle constraints.

Table 2.2: Comparison of benefits and disadvantages in motion planning techniques

Technique	**Advantages**	**Disadvantages**
Line and circle	Low computational cost. Simple to implement. Assures the shortest path for a car-like vehicle, see Figure 2.7a.	The path is not continuous and therefore jerky, making non-comfortable transitions between segments of the path. The planner depends on global waypoints.
Clothoids	Transitions to and from curves are done with a linear change in curvature. Highways and road designs implement these curves. Suitable for local planning (see Figure 2.7b).	Time consuming because of the integrals that define the curve. The curvature is continuous but not smooth (linear behavior). The planner depends on global waypoints.
Polynomials	Low computational cost. Continuous concatenations of curves are possible (Suitable for comfort).	Curves implemented are usually of 4th degree or higher, difficulting the computation of the coefficients to achieve a determined motion state.
Béziers	Low computational cost. Intuitive manipulation of the curve thanks to the control points that define it. Continuous concatenations of curves are possible (Suitable for comfort). See Figure 2.7c.	Loss of malleability when increasing the curve degree, as well as the computation time increases (Thus, more control points have to be evaluated and correctly placed). The planner depends on global waypoints.
Splines	Low computational cost. The result is a general and continuous curvature path controlled by different knots (see Figure 2.7e).	The solution might not be optimal (from the road fitness and curvature minimization point of view) because its result focuses more on achieving continuity between the parts than malleability to fit road constraints.
Function Optimization	Road and ego-vehicle constraints as well as other road users can be easily taken into account (Figure 2.8).	Time consuming since the optimization of the function takes place at each motion state. Therefore, the optimization is stopped at a given time horizon. The planner depends on global waypoints.

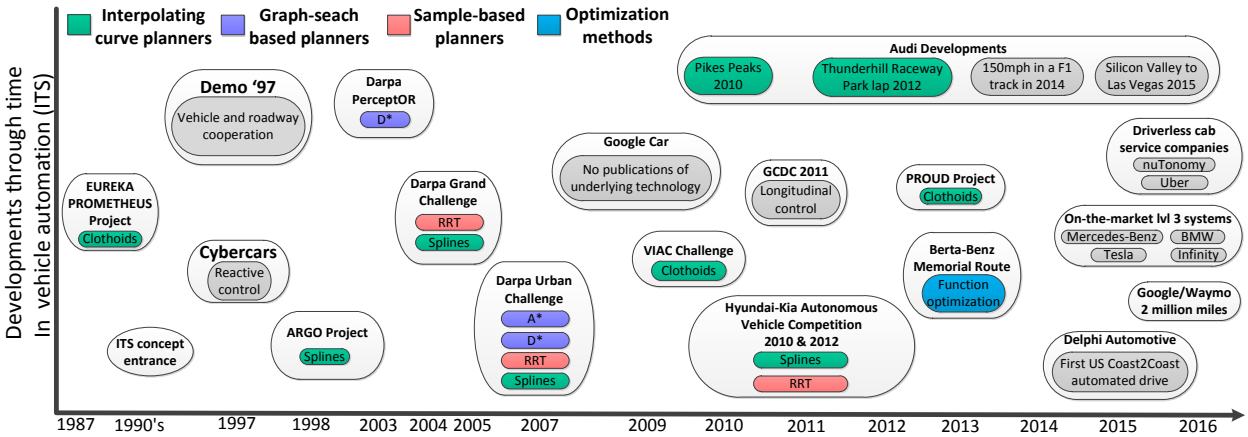


Figure 2.9: Timeline with important automated vehicle demonstrations through time and their associated planning techniques

2.3 Motion planning developments by ITS groups worldwide

Although the idea of intelligent vehicles begun in 1939 at New York world fair with GM's *Futurama* presentation, it took decades to became the idea in a reality. Figure 2.9 shows a timeline with important demonstrations and developments in vehicle automation up to this date[8].. The EUREKA-PROMETHEUS project pionnered in Europe this research between 1987 and 1994, where different vehicles from industrial partners such as Volvo and Daimler were automated. Clothoid paths were implemented in curved segments to aid the control system to follow the route [Behringer and Muller, 1998].

The PATH program presented its platooning demonstration as part of the Demo '97 at San Diego CA, in cooperation with GM [Shladover, 1997]. It consisted on an eight vehicle platoon in a dedicated lane with vehicle inter-distances of $6m$. Carnegie Mellon University participated as well, providing the free agent demonstration with their Navlab vehicles [Thorpe et al., 1998]. Also in the same year, a presentation in The Netherlands introduced the first operational service of CTS [Malone et al., 2002], using shuttle services at Schiphol airport. CTS addressed the last mile problem providing a door-to-door and on-demand service [Parent, 2007]. In the systems above described, reactive control and vehicle-roadway cooperation were preferred over path planning techniques.

One of the first projects to test path planning techniques was the ARGO project from VisLab [Piazzi et al., 2002]. With a vision-based system, the planning technique consisted in adjusting quintic polynomial splines into the lane markings detected by the front camera. Later, the Darpa PerceptOR program stimulated the development of automated navigation techniques for off-road vehicles [Kelly et al., 2006], [Krotkov et al., 2007]. This was achieved by the Darpa Challenges (Grand and Urban challenge) that encouraged the research in automated navigation systems, developing path planning techniques (see Figure 2.9).

The first Google Car made its entrance in 2009. Google driverless cars have carried out several public exhibitions showing the automated capabilities of their vehicles, achieving more than 2 million miles of travel with just one accident attributed to automation[9]. Google also pushed for

[8]Refer here for ITS developments in the U.S.: http://www.its.dot.gov/about/HistoryITS_Timeline.pdf
[9]Available online at: https://www.waymo.com/

the legislation of this kind of technology, receiving the first license for an automated vehicle in May 2012 (two months after the state of Nevada enacted the first automated vehicle law) [Anderson et al., 2014]. Unfortunately there are no publications about the motion planning or control techniques that have been used to achieve those results.

The VisLab International Autonomous Challenge (VIAC) was carried out in 2010 [Bertozzi et al., 2011]. 13.000km traveled from Italy (Parma) to China (Shanghai) in autonomous mode, where the planning consisted on a cost function that considered clothoid curve generation [Broggi et al., 2012]. In the same year and also in 2012, the Hyundai Motor Group organized their own automated vehicle competitions in Korea. The 2010 challenge focused in path following and obstacle avoidance. The 2012 challenge focused in understanding urban driving and detecting traffic signals [Jo et al., 2014]. Motion planning techniques included clothoids [Chu et al., 2012], [Jo et al., 2015], RRT [Jo et al., 2013], among others. Also, the public road urban driverless-car test (PROUD-Car test) was coordinated by Parma University and VisLab, achieving in 2013 the insertion of a vehicle in mixed traffic with nobody on the driver seat[10] (see [Broggi et al., 2014]).

The European Grand Cooperative Driving Challenge (GCDC) in 2011 promoted longitudinal vehicle control [van Nunen et al., 2012]. Nonetheless, some teams have already developed path planning strategies (such as state lattices by AnnieWAY [Geiger et al., 2012]). In this sense, a second call for an urban edition of the GCDC Challenge was held in the Netherlands in 2016[11], where platoon merge, busy road merging and urban environments were considered.

Also, Audi presented high performance vehicle control in a conjoint effort with Stanford [Funke et al., 2012], [Kritayakirana and Gerdes, 2012], implementing clothoids to plan curved segments. Among the demonstrations: 1) the climbe of Pikes Peak in 2010; 2) a racing lap at the Thunderhill Raceway Park in 2012, reaching 150mph in a F1 track in 2014[12]; and 3) automated driving in real scenarios from Silicon Valley to Las Vegas in 2015[13]. Also in 2015, Delphi Automotive made the first coast to coast driving in the US[14], going from San Francisco to New York in 9 days. In Table 2.3 an overview of international research groups is presented.

In 2015, Tesla proposed its "Autopilot" concept. The system consists in the simultaneous implementation of some ADAS such as ACC and LKAS, among others, presenting a level 3 system that allows the vehicle to follow lane markings and adapt its speed to the vehicle in front. Other systems have also been presented by other manufacturers as BMW, Mercedes-Benz and Infinity, reaching a similar behavior, being Tesla the only one capable of automated lane changes if the user presses the turning light button. However, These systems are only designed for highways and peri-urban environments, lacking decision making and motion planning, requesting drivers' attention all the time in case the systems can't handle the situation in-hand.

Recently, two companies joined Google into the testing phase for automated vehicles in urban traffic (level 5 automation). The first was nuTonomy[15], providing the first driverless cab service in Singapore (in partnership with Grab). Test are in initial phase with test-drivers still behind the steering wheel to prevent risky situations. Uber is the second company to accomplish the same thing but in the city of Pittsburgh. Their self-driving cab service is also in test phase, aiming to a complete automation of their rides in 2018. Moreover, the U.S. DoT offered $40 million to Columbus OH after winning the Smart City Challenge. The idea is to define and implement what it means to be a smart city and fully integrate innovative technologies—self-driving cars, connected

[10] Available online at: http://vislab.it/proud-en/
[11] Available online at: http://www.gcdc.net/
[12] Available online at: http://goo.gl/g2eJ50
[13] Available online at: https://goo.gl/voT0cb
[14] Available online at: http://www.delphi.com/delphi-drive
[15] Available online at: http://nutonomy.com/

vehicles, and smart sensors—into their transportation network[16].

Table 2.3: International groups focused in automated vehicles research

Team	Main Focus and Related work
Carnegie Mellon University	**Main Focus:** One of the first robotic center to focus in ITS. Key player of Demo '97 and different Darpa Challenges. Motion planning research is focused in graph search algorithms as D* and state lattices. **Related work:** Demo '97 [Thorpe et al., 1998], DARPA PerceptOR [Kelly et al., 2006], Grand Challenge [Urmson et al., 2006], Urban Challenge [Urmson et al., 2008], [Ferguson et al., 2008], D* [Stentz, 1994], [Ferguson and Stentz, 2006], state lattices [Pivtoraiko and Kelly, 2005], [Howard et al., 2008], [McNaughton et al., 2011], [Gu and Dolan, 2012], [Gu et al., 2013], automated driving platform [Wei et al., 2013], [Gu and Dolan, 2014], [Xu et al., 2014]
Google	**Main Focus:** Automated driving in urban and highway navigation, where many challenges are still unsolved. Their vehicles rely on extensively mapped driveways for motion planning. **Related work:** Publications only by videos or talks, however their results are remarkable enough to be considered as the main private developer of automated vehicles [Anderson et al., 2014].
Hanyang University	**Main Focus:** Winners of the Hyundai Motor Group 2012 Autonomous vehicle competition (their vehicle is called A1). Laser and camera data fusion with spline interpolation as its main motion planning technique. **Related work:** [Chu et al., 2012], [Jo et al., 2013], [Jo et al., 2015], [Jo et al., 2014]
INRIA	**Main Focus:** Research in perception, control, communications and traffic modeling for sustainable mobility. Planning methods focus in polynomial, Bézier curves, clothoids and RRT. Door-to-door and on-demand applications (Cybercars) **Related work:** Clothoids [Fraichard and Scheuer, 2004], RRT [Fulgenzi et al., 2008], [Fulgenzi et al., 2009], Cybercars [Parent, 2007], [Bouraoui et al., 2011], control architecture [González and Perez, 2013], overtaking maneuvers [Petrov and Nashashibi, 2013], [Petrov and Nashashibi, 2014], [Gonzalez et al., 2014],
Karlsruhe Institute of Technology	**Main Focus:** Participants of the Darpa Urban Challenge. Motion planning research begins with graph search algorithms (A* and spatiotemporal lattices) and recent implementations move toward numerical optimization approaches. **Related work:** Behavioral planning [Werling et al., 2008], Darpa Urban Challenge [Ziegler et al., 2008], [Kammel et al., 2008], state lattices [Ziegler and Stiller, 2009], [Werling et al., 2011], cooperation with Stanford [Werling et al., 2010], Grand Cooperative Driving Challenge [Geiger et al., 2012], Berta-Benz Memorial route [Ziegler et al., 2014b]

[16]Available at: https://www.transportation.gov/smartcity

Table 2.3: International groups focused in automated vehicles research

Team	Main Focus and Related work
Massachusetts Institute of Technology	**Main Focus:** Rapidly-exploring Random Tree has been the motion planning technique implemented by MIT since its entry to the Darpa Urban Challenge. Today its knowledge is applied to industrial and safety applications with some spin-offs as nuTonomy. **Related work:** Darpa Urban Challenge [Kuwata et al., 2008], [Kuwata et al., 2009], industrial [Karaman et al., 2011] and safety applications [Anderson et al., 2012], [Aoude et al., 2010].
Parma University and Vislab	**Main Focus:** Focused in perception by vision, this group has been around since the Eureka-Prometheus project. Motion planning techniques include splines (ARGO), RRT (Darpa Challenges), and more recently clothoids (VIAC and PROUD projects) **Related work:** The Argo project [Broggi et al., 1999], [Piazzi et al., 2002], Darpa Grand Challenge [Braid et al., 2006], Urban Challenge [Chen et al., 2008], VIAC project [Broggi et al., 2010], [Broggi et al., 2012], [Bertozzi et al., 2011], future vision [Broggi et al., 2013a], PROUD project [Broggi et al., 2014], [Broggi et al., 2013b]
California Path	**Main Focus:**One of the first ITS research center in the world (focused in automated highways systems), and key participant in Demo '97. Motion planning is mostly infrastructure dependent, with robust control techniques for path following. **Related work:** Demo '97 [Shladover, 1997], [Thorpe et al., 1998], Path at 2007 [Shladover, 2007]
Stanford University	**Main Focus:** From Darpa Grand and Urban challenges to an automated Audi TTS that can drift, is clear Stanford is present in the ITS development. Planning techniques include the A* family, optimization via conjugate gradient and more recently interpolation by clothoids **Related work:** Darpa Grand Challenge [Thrun et al., 2006], Urban Challenge [Montemerlo et al., 2008], [Dolgov et al., 2010], [Kummerle et al., 2009], automated driving platform [Levinson et al., 2011], [Funke et al., 2012], cooperation with other research institutes [Werling et al., 2010], [Dolgov and Thrun, 2009]
OEMs	**Main Focus:** Motion planning for real implementations **Related work:** Toyota [Han et al., 2010], BMW [Aeberhard et al., 2015], [Werling et al., 2011], GMC [Shladover, 1997], [Thorpe et al., 1998], [Urmson et al., 2008], [McNaughton et al., 2011], [Gu et al., 2013], [woo Lee and Litkouhi, 2012], Daimler [Behringer and Muller, 1998], [Keller et al., 2011], [Ziegler et al., 2014b], Volkswagen Group [Thrun et al., 2006], [Montemerlo et al., 2008], [Funke et al., 2012], [Furgale et al., 2013], Hyundai-Kia Group [Jo et al., 2013], [Chu et al., 2012], Renault [Vorobieva et al., 2013], Oshkosh Truck Corp. [Chen et al., 2008], [Braid et al., 2006], Volvo Group [Madas et al., 2013].

2.4 Discussion

Current ADAS developments are reaching a high market penetration, providing a safer and more comfortable driving. These systems do not yet consider decision and motion planning approaches, which are paramount for a full transition towards automated vehicles in the near future. The development of intelligent planning algorithms is one of the important issues when designing robotic architectures [Hwang and Ahuja, 1992], [Elbanhawi and Simic, 2014]. The associated constrains in automated vehicles research—i.e. traffic rules, high speeds or road layout—turn real-time motion planning in a critical aspect for achieving automated driving on complex environments.

This chapter reviewed the motion planning state of the art from two points of view: 1) considering the different techniques (see Table 2.1), there are four main algorithms: graph-search, sample-based, interpolation and numerical optimization; and 2) considering the worldwide research groups that have implemented motion planning techniques on automated vehicles, testing the behavior on real roads (see Table 2.3).

There are two main algorithms that have been recently used real implementations:

- Most of research groups implement interpolation to solve the planning problems. Specifically, recent demonstrations have used clothoids (Audi, Parma/Vislab or Stanford), Bézier curves (INRIA) and polynomial curves (Daimler or INRIA). The main reason for choosing this technique is because an enhanced map in structured environments can provide the needed way-points—i.e. GPS data. Then, optimal curves in terms of smoothness, continuity, vehicle constraints, speed and comfort are generated.

- Graph search is the second algorithm most applied when dealing with real implementations. Specifically, state lattices is the most used one—i.e. CMU, KIT, GMC. The main reason for choosing this technique is because a fast search of the optimal path can be performed in real-time, considering comfort, safety and vehicle constraints although the solution is only resolution complete (depends on the resolution of the underlying grid/lattice).

Current challenges are associated to real-time planning calculation on dynamic environments. Urban scenarios where multiple agents—i.e. pedestrians, cyclists, other vehicles—have to be taken into account, require a continuous evaluation (and re-calculation) of the determined trajectories. The limited time for generating a new free-collision trajectory with multiple dynamic obstacles is an unsolved challenge. This is mainly caused by the time-consuming perception algorithms [Gandhi and Trivedi, 2007], reducing dramatically the motion planning decision window. Current implementations are not able to overcome this limitation yet.

Since planning stage is the link between perception and control, current works considers both: perception uncertainties and control constrains when developing new path planning algorithms. Recent developments in path planning aim to properly handle the uncertainty in the data acquisition process for dynamic environments. This permits to have a better environment understanding in real-time, giving the planning process the ability to prevent dangerous situations by considering uncertainties on the acquisition/perception stages of the architecture. From the control point of view, multiple constrains have to be considered, ranging from vehicle kinematic/dynamic limitations up to passengers' comfort. Recent developments aim to consider some of these constrains in the planning stage leading to smooth and achievable trajectories, reducing control stage constrains. Next steps should be focused on developing algorithms able to fuse perception uncertainties with control constraints. This work has been already started by KIT [Ziegler et al., 2014b] and CMU [Xu et al., 2014].

A new trend on the automated vehicle research is to add the driver in the control loop. From the path planning research perspective, the trajectory is communicated to the driver using an HMI, acting as an ADAS [Glaser et al., 2010], [Hardy and Campbell, 2013]. This creates a new research challenge by multi-fusing perception uncertainties, control constrains and driver knowledge when generating safe, smooth and achievable trajectories.

Chapitre

Planification des trajectoires pour des environnements statiques

Below is a French summary of the following chapter "Trajectory planning in static environments".

Les approches proposées dans la littérature ne permettent pas de générer automatiquement une trajectoire en temps réel d'un véhicule dans un environnement urbain ou semi-urbain. Le principal défi est la gestion des obstacles dynamiques auquel le véhicule doit faire face. La sécurité, le confort et le temps de déplacement des passagers peuvent se traduire en terme de qualité des trajectoires à créer (continues, lisses, etc.). Dans cette thèse, une nouvelle architecture fonctionnelle pour la planification automatisée des trajectoires de véhicules est introduite, constituée de deux composantes principales :

- **Planification globale :** à partir du point de départ du véhicule, du point d'arrivée et d'une carte numérique, un ensemble de points décrivant grossièrement le chemin à suivre est généré. Chaque point est associé à des informations concernant l'environnement statique, comme la disposition de la route, la vitesse maximale autorisée, les informations d'intersection, etc.
- **Planification locale :** à partir des points donnés par la planification globale, une trajectoire est générée en temps réel. Son support est une courbe de Bézier qui tient compte des contraintes liées à l'environnement et au véhicule. Afin de réduire les accélérations générales et augmenter le confort des passagers, on cherche à minimiser la courbure du chemin. Puis les profils de vitesse sont calculés en tenant compte des contraintes cinématiques et dynamiques du véhicule.

Des environnements complexes comme les ronds-points ont également été étudiés. Dans ce cas, une approche nécessitant la résolution de trois sous-problèmes est proposé : l'entrée et la sortie du rond-point et le rond-point proprement-dit. Une fois les courbes paramétriques construites, la trajectoire est ensuite donnée à l'étape de contrôle, qui suivra les profils latéraux (chemin physique) et longitudinaux (vitesse). Notre approche permet de résoudre la plupart des situations rencontrées en milieu urbain en l'absence d'obstacles.

Chapter 3

Trajectory planning in static environments

After the state of the art review, is clear that a system capable of generating trajectories in real-time for automated vehicles motion planning is missing. The main challenge is the dynamic environment the vehicle has to face, mostly in urban areas. On the other hand, the continuity and smoothness of the generated trajectories plays a key role, being directly related to passengers safety, comfort and traveling time. Having this in mind, a novel functional architecture for automated vehicles trajectory planning has been designed, based on two main components as follows:

- **Global planning:** Singular waypoints that describe the road characteristics (as its layout, maximum speed, intersection information, etc.) are generated throughout the trajectory the vehicle will follow. The waypoints model the static environment based on a digital map from previous human traveled roads.
- **Local planning:** The singular waypoints are smoothed implementing parametric curves. The idea is to find trajectories in real-time to reduce overall accelerations and increase passenger comfort. First a G^1 continuous path is generated, searching for a minimal curvature profile. Then speed profiles are computed with respect to the curvature of the path, considering the vehicle's kinematic and dynamic constraints.

Complex environments as roundabouts have also been studied, where the proposed solution varies slightly from turns, implementing a three stage approach that adapts parametric curves to this singular intersection. The given trajectory is then passed to the control stage, which will follow the lateral (physical path) and longitudinal (speed) profiles. The novel architecture is general enough to cope with most situations encountered in urban environments and will be validated in simulated and real platforms within the RITS team.

The chapter is structured as follows: Section 3.1 presents the problem description in motion planning based on the state-of-the-art review explained in the previous chapter. The proposed architecture is presented in Section 3.2 describing six main stages from data acquisition to actuation. The proposed global planning strategy is presented in Section 3.3, describing the path and road characteristics (with singular points) for the vehicle to follow. The local planning is based on these singular globar points and is presented in Section 3.4. The main contribution is a novel trajectory planning architecture for the motion planning problem in automated vehicles. It will be able to address a wide range of urban driving situations, planning also a curvature based speed profile to increase passenger comfort. Roundabouts are also considered and explained as a special

intersection in Section 3.5. This circular-shaped intersection is modeled in three steps (entrance, circulatory roadway and exit) to ensure curvature continuity and comfort. Finally, conclusions are presented in Section 3.6

3.1 Problem description

Despite of important developments in automated vehicles technology, lateral actions and strategies remain mostly as warning systems for lane keeping because even small lateral actions over the steering wheel can have dangerous consequences. Today, one of the unsolved challenges in the automated vehicles research field is real-time motion planning, specially in urban environments where many actors are involved, i.e. other vehicles, pedestrians, cyclists, among others.

Some authors have implemented spatio-temporal lattices to evaluate simultaneously path and speed planning. [Ziegler and Stiller, 2009] describe the generation of candidate piecewise trajectories considering position, velocity, acceleration and time (seven dimensions). [McNaughton et al., 2011] presents a similar technique, arguing that the algorithm has a high computational cost and therefore implementing a graphic processing unit (GPU) to reduce this computation time. Moreover, in both approaches, the search space has to be pruned in order to arrive to a solution in time, introducing sub-optimality. On the other hand, RRTs have also been implemented to plan simultaneously a path and a speed profile. In [Jeon et al., 2013] RRT* is implemented as an on-line method, considering limit accelerations for the vehicle. This provides a solution that is somehow continuous but not smooth, making it jerky and causing discomfort.

Demonstrations from Daimler [Ziegler et al., 2014a] and VisLab/Parma university [Broggi et al., 2013a] show important developments in the field, where the system was tested in different real scenarios showing good results. However [Ziegler et al., 2014a] shows no details about overall accelerations/speed suffered by the vehicle and state that the method might not achieve the optimal solution if the planner is stopped (a limit is set at 0.5s per numerical optimization). [Broggi et al., 2013a] implements a clothoid database as a path planning resource then followed by a Model Predictive Controller (MPC) to change the path into lateral steering actions. However no details are given on the speed profile generation to account for vehicle dynamic limits (only relationships between curve databases and speed profiles) or passengers comfort (no evaluation of lateral accelerations). In spite of the remarkable real demonstration carried out, passengers' comfort was not taken into account when generating vehicle trajectories.

Comfort considerations in path and speed planning are found in [Labakhua et al., 2008]. Firstly, the path is determined by splines or clothoids and then the speed profile is computed following the comfort constraints in ISO 2631-1 standard [ISO, 1997]. The result is a continuous but not smooth speed profile that complies with lateral and longitudinal acceleration constrains. [Villagra et al., 2012] also based their approach on lateral comfort accelerations from [ISO, 1997], implementing a jerk and acceleration bounded speed planner over a clothoid based interpolating path planner. This allowed to have lateral and longitudinal accelerations constrained to comfort parameters.The speed profile generator is able to plan the correct speed for each curve and then compare the results with human drivers. This approach is based on planning smooth speed changes outside the curves with the jerk limitation method [Liu, 2002] and limiting target velocities to respect lateral accelerations when curves occur. However, both of the methods were off-line (the planner has to compute all the path at once), keeping longitudinal accelerations out of curved segments (overall velocity in curved segments remains the same throughout the curve) and acceleration profiles were not smooth.

Having this in mind, the current chapter presents a novel local trajectory planner based on

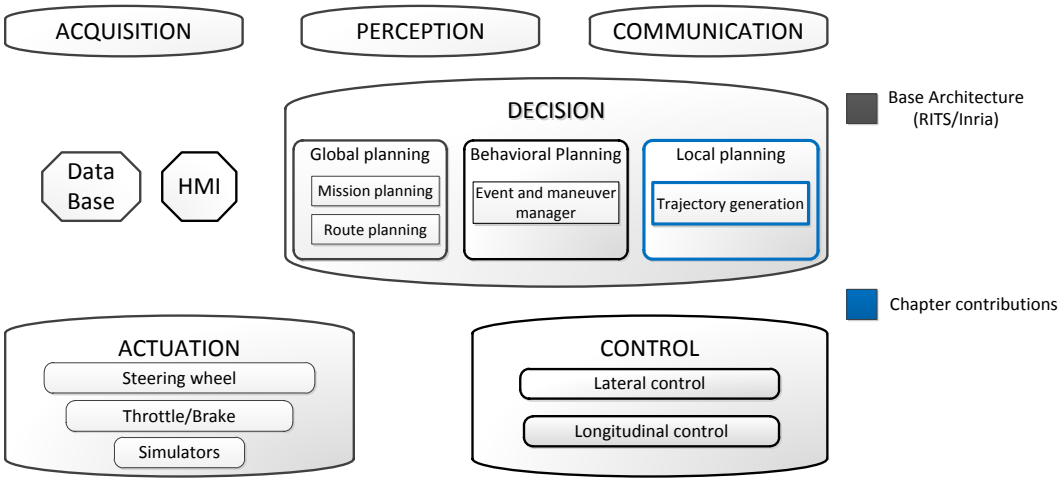


Figure 3.1: RITS team/project architecture. Contributions in the present chapter are shown in blue.

Bézier curves. The main focus is to introduce comfort in a real-time trajectory generation. The planner is capable to create G^1 continuous paths in real-time based on global waypoints information, accounting for vehicle kinematic/dynamic and road layout constraints. G^2 continuous speed profiles are computed on top of the physical path to create optimal trajectories considering passengers' comfort. In the following, the author explains the functional planning architecture and how it fits RITS team/project global architecture.

3.2 RITS Architecture

The RITS team/project is a multidisciplinary research program, working on robotics for ITS. It is part of INRIA of Paris (from french: *"INRIA de Paris"*) and seeks to combine the mathematical tools and techniques to design advanced intelligent robotics systems for automated vehicles and sustainable mobility. The goal of the team is to improve road transportation in terms of safety, efficiency, comfort and also to minimize nuisances. The technical approach is based on drivers aids (ADAS), going all the way to full driving automation.

Figure 3.1 shows RITS team/project architecture for automated vehicles. It is based on architectures previously proposed in the literature (see Figure 2.4). It is composed of six main stages, enabling the vehicle to locate itself in the environment, as well as perceive obstacles or other road users, plan motion trajectories and follow those by applying control strategies.

Acquisition, perception and communication: These first stages collect all sensor data and process it to obtain the state of the vehicle (position, orientation, speed, among others) or the state of the environment (objects, other road users, etc.). To localize the vehicle, two main strategies are implemented: 1) Real Time Kinematic Differential Global Positioning System (RTK-GPS) fused with an Inertial Measurement Unit (IMU) gives the global position in cartesian coordinates; and 2) Simultaneous Localization and Mapping is also used for locations when the RTK-GPS is unavailable [Trehard et al., 2014]. Obstacles are detected and classified from 2D laser information coming from the frontal and back part of the vehicle (see [Merdrignac et al., 2015]). Communications are used to get information from other road users or the infrastructure (e.g. a wireless mesh network over Wi-Fi for cooperative maneuvers [Milanés et al., 2011]).

Decision: Taking the information from perception and communications, this stage is in charge of providing the motion strategies to be followed by the vehicle. The focus of the Ph.D. work lies within this stage to design and develop a motion planning architecture applied to automated vehicles. It is divided in three main blocks: 1) The global planner; 2) The local planner; and 3) The behavioral planner. The global planner gives the global mission for the vehicle to go from point "A" to point "B" in a map. The local planner smooths the global path and provides an obstacle free trajectory navigable by the vehicle. The proposed local planning is composed of two main stages. First, the generation of a local path is performed, being the main goal to minimize path's curvature. Then, a speed profile planning stage generates the speed reference for the vehicle to follow in direct relation with the geometric path. Lateral, longitudinal and total accelerations (euclidean sum of lateral and longitudinal accelerations) limit the speed profile, seeking comfort based on ISO 2631-1 [ISO, 1997]. The behavioral planner: Although it remains out of the scope of this work, it is important to mention it for future works, focusing on the modularity and scalability of the proposed motion planning architecture. In the literature is in charge of managing the current maneuver/strategy to execute, for details please see [Wei et al., 2013].

Control and actuators: These will execute the trajectories planned in the decision stage. The control stage is divided in longitudinal and lateral control. The longitudinal control implements a classical PD controller to follow the smooth speed profile provided by the decision stage, giving throttle and brake actions to the low level system through Controller Area Network (CAN) protocol. The lateral control implements a pure persuit controller to follow the geometrical path given by the planner, sending steering commands to the low level actuators (also through CAN protocol) as in [González and Perez, 2013] and [Roldao et al., 2015].

Data base and HMI: Databases and HMIs are also included as part of the architecture. The former provides necessary information for the vehicle, such as sensor setup, digital map information, etc. The latter allows passengers and drivers to interact with the vehicle and get a feedback on the performance of the embedded system.

At RITS team, early steps towards a functional motion planning architecture have already been set [Bouraoui et al., 2011]. The three month INRIA demonstration at La Rochelle gives an idea of the initial capabilities of the automated system. Despite great advances in laser localization as Simultaneous Localization and Mapping (SLAM) and the "on-demand/door-to-door" system, the decision stage was based on pre-recorded hand made paths. It was useful since the Cybus (cybercar involved in the demo) did not need to change its route. Polynomial curves where defined to take the vehicle closer to each pick-up station instead of real-time trajectory generation, limiting the system and preventing scaling (no local planner existed, only global nodes). The present Ph.D. work aims to provide this missing link proposing a functional motion planning architecture. In the following, the decision sub-stages (global and local planning) will be explained in detail.

3.3 Global Planning

In contrast to path planning algorithms in robotics, where the environment is often unstructured or semi-structured, automated vehicles have to deal with structured-space constraints. In the literature, the approach is often divided in two: global and local planners.

The ability to plan global itineraries received important attention in the early 2000s, with TomTom and Garmin firsts portable navigation devices (2004 and 2005 respectively), followed by turn-by-turn mobile guidance apps like Google Maps (2008), Waze (around 2010) and Apple Maps (2012), thanks to GPS equipped mobile phones. The underlying technology takes user's location

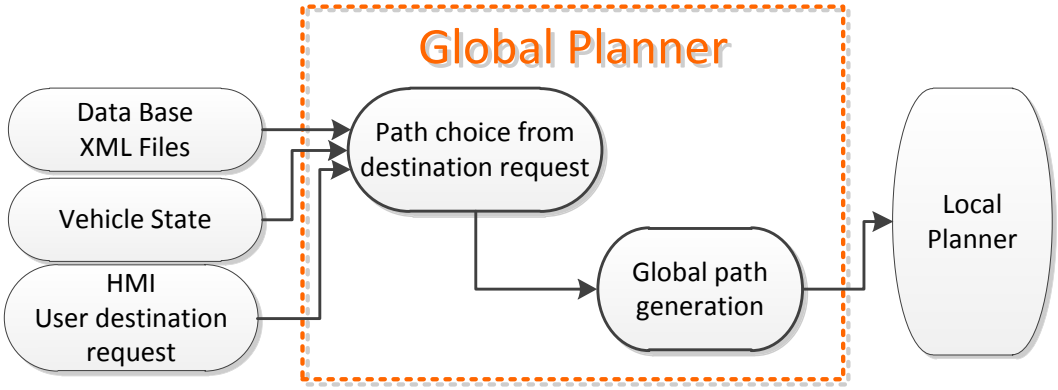


Figure 3.2: Global planning Flowchart.

information, real-time traffic information, digital map road segments and updates, among others, to offer a global path with the least estimated time of arrival, implementing graph-search techniques like A* or Dijkstra's algorithm [Li et al., 2009]. However, those are closed implementations, meaning that the given path is user oriented only and other applications cannot profit from the GPS data used to identify the optimal path.

Approaches for the DARPA urban challenge (2007) as [Montemerlo et al., 2008], [Bohren et al., 2008] and [Urmson et al., 2008] implemented road network definition files from DARPA officials that described road stretches (length, width, waypoints, etc). Then, it sufficed to do a node search to find the least time consuming path according to the mission assigned. [Bertozzi et al., 2011] describes how the Parma-Shanghai travel did not have a well described map for the whole trip, so the global path consisted mostly of lane-keeping, waypoints following or leader following mode. [Xu et al., 2012] and [Gu et al., 2013] relay on detailed global paths that define the lane center, its orientation and curvature. However, is not clear how the process of detailing the path has been done. This section presents the proposed global planning approach. Singular points describe the road shape, intersections and speed associated to each stretch. The main contributions of this section are:

- XML route networks describe navigable paths for the ego-vehicle.
- Determine the route with least estimated time of arrival for the passenger's destination.
- Real-time re-planning of the route if a new destination is set.

3.3.1 Global waypoints route planner

INRIA global planner profits from human knowledge since the waypoints are design by hand taking into account previous human driven routes. These are described as global coordinates x,y points that describe the road (specifically the road shape, maximum speed, turns and roundabouts). In this work the points are provided as an Extensive Markup Language (XML) file. It allows to have the intersection information such as maximum speed of the segment, roundabout radii, entry and exit angles, among others. The information provided by the XML file is as follows:

- Identification (id): Is the identification number of each point in the XML file.
- Cartesian coordinates (X, Y): Gives the global position of the point in a map, with respect to a fixed point defined by the technique used to locate the vehicle as SLAM or GPS.

The information is given in meters. In roundabout shaped intersections, this information represents the middle of the roundabout.

- Speed: Gives the maximum speed of the current stretch in meters per second.
- Type: Defines the type of the points. Since these singular points define the shape of the road, we have two types: 1) Turns (-2): This type describes intersections, but also describes roads non entirely straight. The idea is to be able to address as much road shapes as possibles and in urban environments these points describe the street shape in all situations; and 2) Roundabouts (-1): This type is necessary since roundabouts present unique road characteristics that are difficult to be addressed by only implementing turn type points. When describing roundabouts, three more informations are needed as follows:
- Radius: This information describes the external radius of the roundabout in meters.
- Angle in and angle out: Define the entry and exit angles from the roundabout. These permit to define roundabouts that have incoming roads non-aligned with its center by describing the entry and exit angle.

The structure of the XML file is done as follows:

```
<network>
  <link>
    <node id="1" x="204.4" y="5.2" speed="3" ></node>
    <node id="2" x="222.04" y="5.82" speed="3" type="-2" ></node>
    <node id="3" x="238.8" y="15.4" speed="3" type="-1" radius="7.5" angleIn="17.5" angleOut="31.5"
></node>
    <node id="4" x="241.02" y="28.24" speed="3" type="-2" ></node>
    <node id="5" x="213.5" y="56.0" speed="3" ></node>
  </link>
</network>
```

where the network is defined as a set of links representing each of the stretches with the information in every node.

Figure 3.2 describes the global planning approach. The flowchart shows the core behavior of the component, and receives information from the database, the vehicle state (localization, velocity, among other information from the ego-vehicle proprioceptive sensors) and the HMI (essentially the destination request). It first reads the XML file with globally referenced points and the description of the intersection they represent. Then the complete global path is created from these points adding inner roundabout points (middle points). All the information of the road layout is included in the databases. The component waits for a destination to arrive (X, Y format) and when is received, the component limits the global path points to only those in between the current position of the vehicle and the chosen destination point. The reception of the destination vector is implemented as a trigger to the whole system, only sending the global points to the local planner when the destination points are set by the user.

Once the map is computed, it is sent to the local planning stage. In case a new destination is requested, the global planner can re-plan the path in real-time. The planner takes into account the current position of the vehicle, the new requested destination and the final destination to produce a new path (see Figure 3.3). Notice the new path in red, the previous path in blue and the new destination requested as "intermadiate point". Notice also that the intermediate point is reached

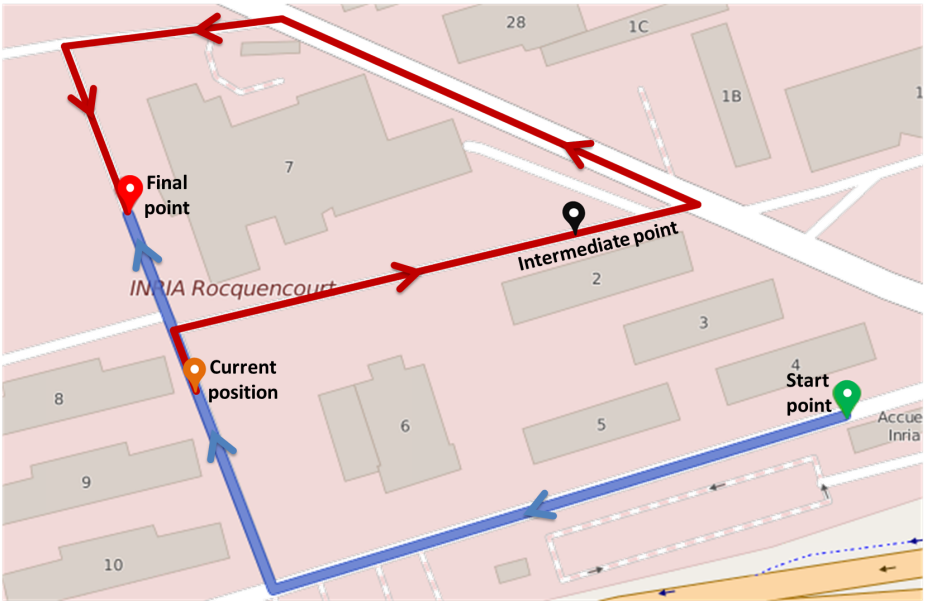


Figure 3.3: Global planning at INRIA facilities (image from OSRM project).

earlier because is declared with a higher priority. The presented approach allows to create paths that can be implemented as bus itineraries that include door-to-door and on-demand concepts as the Cybernetic Transportation Systems (CTS) [Bouraoui et al., 2011], [Vaca et al., 2016].

3.4 Local planning

The global planner provides singular points in the trajectory. However, this set of waypoints is not enough for achieving a good performance from an autonomous driving perspective. The local planner is then in charge of smoothing the path and make it navigable for the vehicle. Figure 3.4 presents the local planning related modules proposed in this Ph.D. work. The information coming from the global path (road layout information) and the vehicle state (localization, velocity, among other information from the ego-vehicle) are taken into account to produce a comfortable path. The planner is divided in two main modules: 1) Geometric path generation (generation of x, y points to be followed by the vehicle); and 2) Speed profile generation (generation of speed profiles with respect to covered distance within the path).

The final output of the local planner is a trajectory composed of a guidance path for the vehicle, that will translate to lateral actions in the control stage of the architecture (geometric path) and a speed profile that describes the longitudinal actions. The local planner algorithm is designed to deal with any road configuration, being the automated vehicle able to navigate in any urban environment, describing minimal curvature and continuous paths within a receding horizon. Once the physical path is described, speed profiles are generated over this first path producing the final trajectory to be followed by the vehicle. The speed profile will be constrained by comfort accelerations defined in ISO 2631-1 [ISO, 1997], creating a trajectory within the limits of comfort. The main contributions of this section are:

- Real-time G^1 continuous path for urban scenarios using parametric curves.
- Minimal curvature paths, reducing lateral accelerations and increasing comfort.

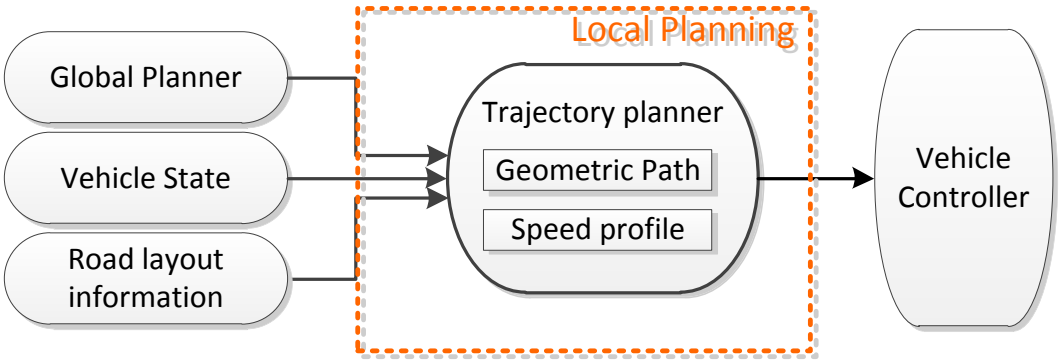


Figure 3.4: Local planning related modules for static environments.

- Real-time speed profile generation, complying with comfort constraints in [ISO, 1997].

3.4.1 Bezier curves as a basis for trajectory planning

When it comes to design comfort-based motion planning algorithms for automated vehicles, trajectory generation is directly linked to the continuity of both, the path and the speed profile, keeping smooth changes between different motion states and avoiding jerky behaviors. Chapter 2 presented the current state of the art which included four main planning techniques (graph-based planners, sample-based planners, interpolating curve planners and numerical optimization). Graph-based planners heavily rely on grid search, where the environment has been previously discretized (most of the times the optimal solution needs a smoothing phase to cope with grid discontinuities [Montemerlo et al., 2008]). Sample-based planners can give a proper answer in time but the continuity is not ensured, losing smoothness and increasing jerky profiles. Numerical optimization approaches can be continuous but the optimal solution is not ensured and the planner is computationally costly. Having this in mind, the interpolating curve planners where chosen because of the compatibility with RITS architecture (global path defined with waypoints) and planning goals (fast trajectory generation; kinematic, dynamic and comfort constraints can be included by implementing continuous curve profiles).

Within the interpolating curve planners family, five different curve design strategies are presented and compared in Table 2.2, describing the pros and cons between them. These are as follows:

- Line and circle is the least computationally expensive of them, however, joints are discontinuous creating jerky paths.
- Clothoids can achieve C^2 continuity at joints but are computationally expensive because of the integrative basis and lack flexibility.
- Polynomials are capable of providing continuous paths (C^n continuity), however degree elevation is needed for path continuity, which complicates the coefficients computation.
- Splines give a good compromise between computation time and path continuity, however, continuity constraints at knots prevents, in some cases, the adaptability of the curve.
- Bézier curves have a good compromise between computation time and path continuity and their adaptability/flexibility is higher (depending only of control points), where the positioning of the control points is intuitive (even when the curve degree is elevated).

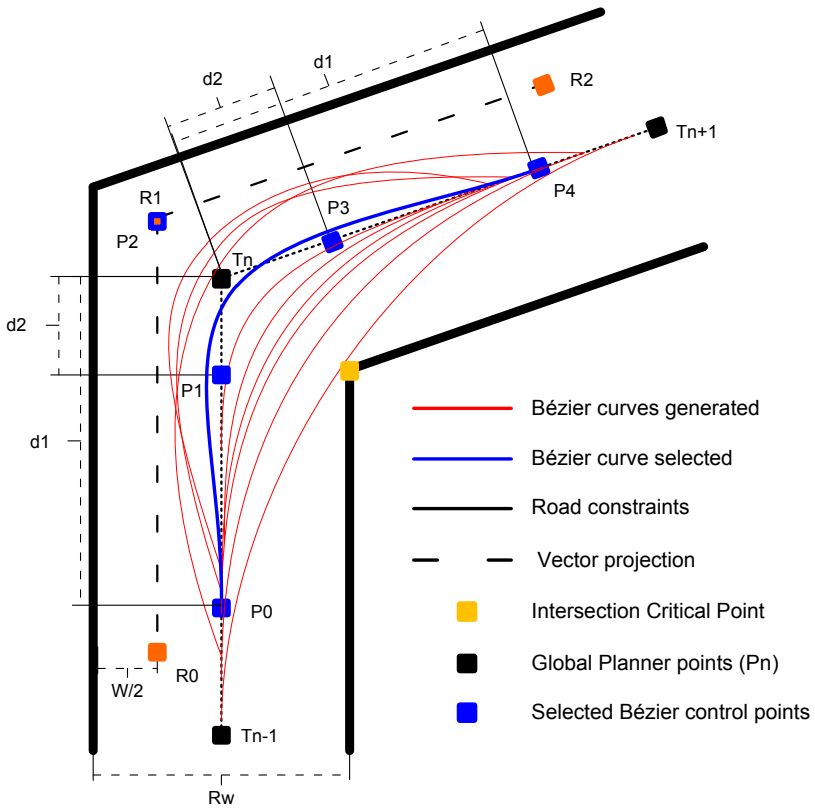


Figure 3.5: Intelligent intersection curve generation

The strategy chosen to design the functional planning architecture are Bézier curves because of the good tradeoff between computational time, flexibility and intuitive control points position. It is mathematically described by Equation 3.1,

$$\mathbf{B}(t) = \sum_{i=0}^{n} P_i \binom{n}{i} (1-t)^{n-i} t^i \quad t \in [0,1] \tag{3.1}$$

where n is the degree of the polynomial equation and P_i are the control points that define the curve. The t parameter is defined between [0,1]; an iterative process generates the curve at each entry or exit stage. Bézier curves have several properties that aid in the design of the optimization algorithm to find continuous and minimal curvature paths. Some important ones are as follows:

- The curve begins at P_0 with direction defined by $\overrightarrow{P_0P_1}$ and ends at P_n with direction $\overrightarrow{P_{n-1}P_n}$
- The control points define the convex hull where the curve is defined.
- Bézier curve possess the variation diminishing property. Meaning that the number of inflection points are equal or less than the polygon formed by the control points and naturally smoothing the stretch they cover.

3.4.2 Singular points handling

The global planner provides singular points which describe the road characteristics. However, these are not smooth since they present sharp corners, making transitions uncomfortable between each of these global points (even at small speeds). The local planner addresses these as turning stretches and intersections with Bézier curves. The global path is smoothed with continuous parametric curves, aiming for G^1 continuity and the minimization of the overall curvature in the path. This reduces lateral accelerations, while considering at the same time the road constraints (the path stays within limits, avoiding curbs) and the vehicle limitations (as the vehicle width—W, the maximum vehicle turning radius, etc). Figure 3.5 presents the generation of Bézier curves (red and blue) with respect to the singular points (in black).

To achieve a continuous path profile, a weighting algorithm has been implemented. Its computation gives the best parametric curve to be set in the region of the singular point, achieving G^1 continuity between straight and curved segments and minimizing curvature gaps. The proposed approach generates several control points configuration, evaluating the curvature profiles at the beginning and end of the stretch (giving each of them a weight, seeking curvature continuity). It also compares the maximum curvature described by the curve with the physical maximum of the vehicle (it discards the ones physically impossible to achieve). Moreover, since each curve must comply with the road layout, the control points will always be generated within the road limits to profit from the convex hull property of Bézier curves (forcing the curve to always comply with the outside boundary) and comparing the set of curves only with the inner critical point of the curb, discarding those exceeding the road boundaries. 3rd degree Béziers are generated as a first resource. If the curve does not meet the requirements established, $4th$ degree Béziers are implemented where the new control point will serve to pull the curve away from the inner curb (see Figure 3.5). The curve evaluation is performed as follows:

- The points T_{n-1}, T_n and T_{n+1} describe the raw path points coming from the global planning, where the control points will be set (i.e. P_0, P_1, P_3, and P_4). This implementation ensures that the curve will respect the external bounds, relying on the convex hull property (see Figure 3.5).

- The vectors $\overrightarrow{R_0R_1}$ and $\overrightarrow{R_1R_2}$ limit the external band of the road where the external control point will be set to pull the curve away from the critical curb point (in case 4th degree Béziers are needed, i.e. P_2). The limit is set at half the vehicle width ($W/2$) of separation with respect to the road boundaries. This generates the control points of the Bézier in a way that its convex hull complies with the external road boundaries of the intersection (see Figure 3.5), forcing the curve to also fit the external road constraints.

- The maximum curvature of each Bézier curve must be less than the physical turning limits of the vehicle (see Equation 3.8). Otherwise, the curve is discarded.

- The generated curves must fit the internal road constraints, i.e. all the curves have to be at least half vehicle width ($W/2$) away from road boundaries (R_w). Special attention is given to the intersection critical point (yellow point in Figure 3.5).

The location of the control points for Bézier curve is described by the following equations:

$$P_0 = T_n + d_1 \frac{T_{n-1} - T_n}{\|T_{n-1} - T_n\|} \tag{3.2}$$

$$P_1 = T_n + d_2 \frac{T_{n-1} - T_n}{\|T_{n-1} - T_n\|} \tag{3.3}$$

$$P_2 = T_n + d_3 \frac{R_1 - T_n}{\|R_1 - T_n\|} \tag{3.4}$$

$$P_3 = T_n + d_2 \frac{T_{n+1} - T_n}{\|T_{n+1} - T_n\|} \tag{3.5}$$

$$P_4 = T_n + d_1 \frac{T_{n+1} - T_n}{\|T_{n+1} - T_n\|} \tag{3.6}$$

where d_1 is the distance from T_n to P_0 and P_4; and d_2 is the distance from T_n to P_1 and P_3 (see Figure 3.5). 3rd degree Béziers are only described by P_0, P_1 P_3 and P_4. The point P_2 will be placed within the vector described by $\overrightarrow{T_n, R_1}$ at a distance d_3 to create 4th degree Béziers in case is needed. This pulls the curve away from the inner intersection critical point.

$$k(t) = \frac{\mathbf{B}'(t) \times \mathbf{B}''(t)}{||\mathbf{B}'(t)||^3} \tag{3.7}$$

$$k_{max} = \frac{\tan(\phi_{max})}{Wb_{ego}} \tag{3.8}$$

Equation 3.7 describes the curvature $k_{(t)}$ of a parametric curve as in [Walton et al., 2003]. It computes the curvature of $B_{(t)}$ (see, Equation 3.1), being $B'(t)$ and $B''(t)$ the first and second derivative of the Bézier curve $B_{(t)}$. Equation 3.8 represents the maximum curvature a vehicle describes, when the turning angle of the wheels attains it maximum value, i.e. ϕ_{max}, with respect to the vehicle wheelbase (Wb_{ego}).

3.4.2a Adjacent consecutive intersections

The presented trajectory generation is able to plan turns and intersections from global singular points. However, it was limited by the amount of space needed between two different consecutive waypoints to generate a continuous path. These closely placed global points are a common situation in urban scenarios (especially at downtown), where the roads are not homogenous (either the road is curvy and needs to be modeled with several points, or there are several intersections in a row). A suitable approach for adjacent consecutive singular points is proposed in this section.

The limiting problem is due to the consecutive joint points of two (or more) curves when the continuity of the path needs to be ensured. In previous works [Yang and Sukkarieh, 2010, Walton et al., 2003], a study on the curvature continuity for Bézier and parametric curves has been applied by considering the derivative of the curve at the junction point. The study of the Bézier properties permits to find the best place to set the control points in order to have C^1 (parametric) or G^1 (geometric) continuity. To achieve C^1, it suffices to ensure that the ending tangent vector of the first curve is identical to the first tangent vector of the second curve, as in Equation 3.9.

$$\vec{Q'_{(1)}} = m(Q_m - Q_{m-1}) = \vec{R'_{(0)}} = n(R_1 - R_0) \tag{3.9}$$

where $Q_{(t)}$ and $R_{(t)}$ are m and n degree Bézier curves respectively. $Q_0, Q_1, ..., Q_m$ are the control points of $Q_{(t)}$ and $R_0, R_1, ..., R_n$ are the control points of $R_{(t)}$.

Figure 3.6 shows three adjacent Bézier curves. C^1 continuity could allow the path to be parametric continuous for the first derivative of both joint curves, which is needed to have a smooth behavior when the vehicle navigates the path. However, from Equation 3.9 is clear that

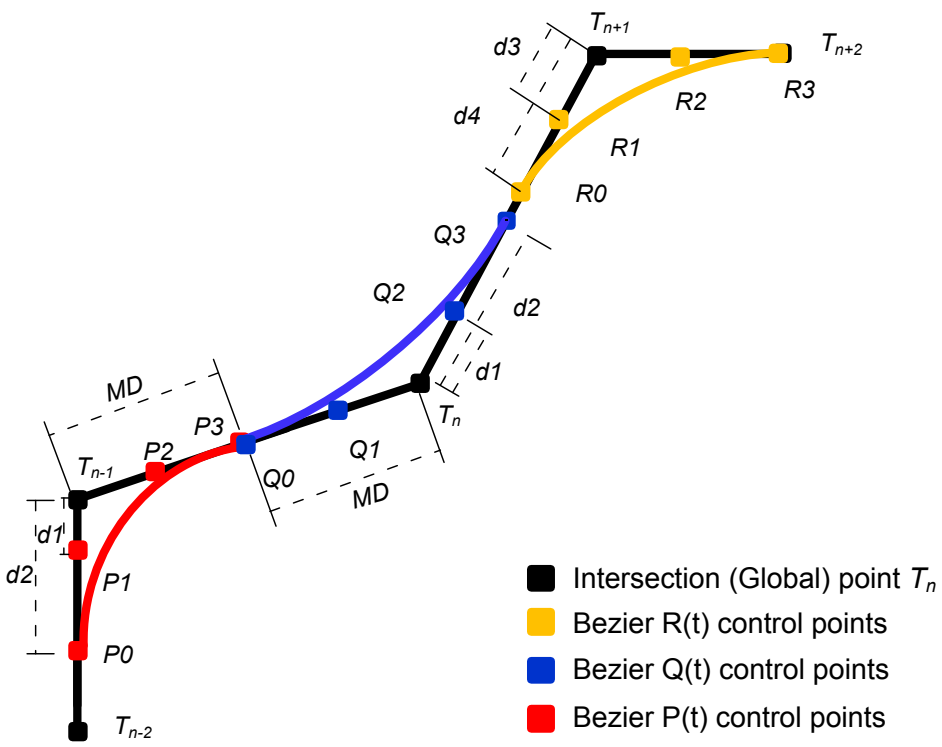


Figure 3.6: Solution for consecutive intersections

the condition is restricting the position of control points in joint curves and therefore restricting the shape of following curves. In order to increase path generation flexibility (i.e. be able to manipulate the curve and adapt to the structured constraints of the road), G^1 geometric continuity is therefore applied. This is a less-restrictive condition while still providing tangent path continuity. It means that control points are only restricted to respect tangent continuity and not necessarily the parametric one (it profits from the first Bézier property presented).

In order to achieve G^1 continuity in the path, the local planner takes into account the space between each intersection point (T_n). First, the intersection distances $\overrightarrow{T_{n-1}T_n}$ and $\overrightarrow{T_nT_{n+1}}$ are calculated (see Figure 3.6); if the control points maximum distance (MD) set in the planner is bigger than half the minor intersection distance—i.e. min $((T_{n-1}T_n)/2$, $(T_nT_{n+1})/2)$—a junction is required. Therefore, minimum distance is MD. Figure 3.6 shows when $Q_{(t)}$ (blue curve) needs to adjust to $P_{(t)}$ (red curve) and $R_{(t)}$ (yellow curve), being $(T_{n-1}T_n)/2$ the constraint distance for the Bézier generation. Once the minimum distance is defined, the computation of the Bézier curve is done as in Section 3.4.2.

Algorithm 1 presents the implementation proposed with Bézier curves at turns and intersections. Lines one to three set the global information for the local planner. The generation and evaluation of Bézier curves is computed from lines four to 21. 3rd degree Béziers are computed first, iterating through distances d_1 and d_2, setting control points in line six. The best curve will be the one minimizing the addition of weights assigned in line 12, excluding those not complying with line 11. If a 3rd degree curve is found then stop the search, generate the curve and send it downstream to the control stage. In case no curve is found (i.e. not fulfilling line 18 statement), 4th degree Béziers are generated. To achieve this, P_2 is generated and the process is repeated. Finally if no navigable curves are found through the whole process, the algorithm request a new

Algorithm 1 Path generation at intersections

```
Start
Read algorithm properties
Get geometrical data of the route—global path, road width, etc.
for (n=3 ; n≤ 4 ; n++) do
    while curve evaluation process is not completed do
        Set control points P_0, P_1, P_3 and P_4
        if n == 4 then
            Set control point P_2
        end if
        Evaluate k_start, k_end and k_max from each curve
        if  k_max is ≤ vehicle's maximum feasible k then
            Assign weights to max(k_start, k_end) and k_max
            Save control points and their associated weights
        else discard the curve
        end if
    end while
    Evaluate all the curves choosing the one with the least weight
    if A curve has been found then
        Break the For loop and provide the optimal curve
    end if
end for
if No curve was found then
    Inform control stage for safety actions
    Inform decision stage requesting new path strategy
end if
Generate the best curve (iterate through t)
Send the path through the communication buffer
End
```

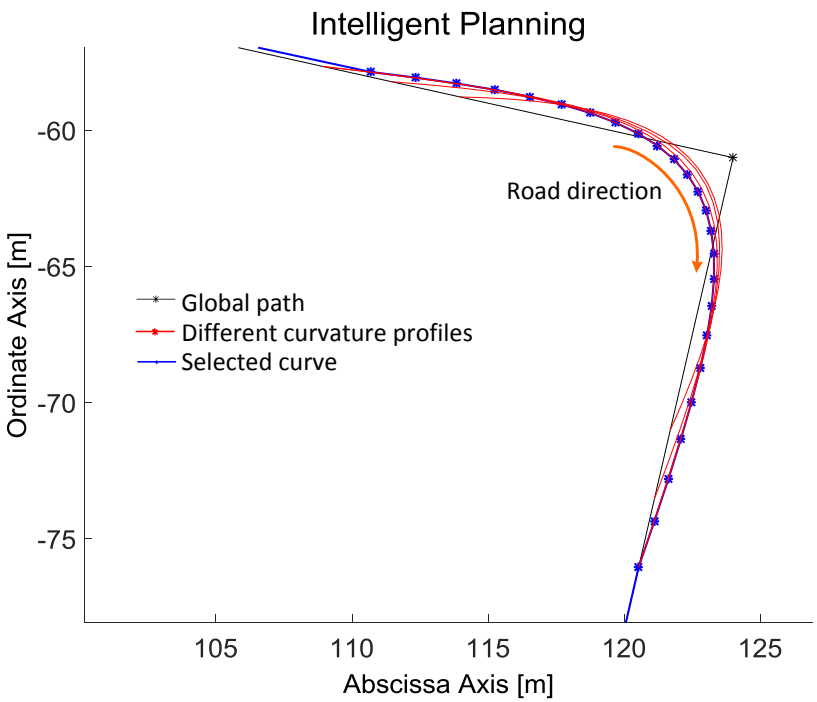


Figure 3.7: Planning different curves to find the best curvature profile

strategy from the global planner, and notifies the control stage to take safety measures (lines 23 and 24).

Figure 3.7 depicts the intelligent planning of an intersection or turn (smoothing singular points). Different curvature profiles are calculated (red lines), evaluated as in Algorithm 1. The one selected as the most suitable curve to apply is represented in blue (having the least curvature profile among the others and complying with road constraints).

3.4.3 Speed planning and trajectory generation

Motion planning research on automated vehicles has been mainly focused on generating optimal paths with respect to length, curvature, its derivative, or a combination of them optimized by a cost function as in [Ziegler et al., 2014b] or [Gu and Dolan, 2012]. However, the planning of a proper speed profile has received little attention, as well as the consideration of lateral and longitudinal accelerations and jerk for passenger comfort [Villagra et al., 2012].

Some authors have implemented spatio-temporal lattices to evaluate simultaneously path and speed planning. Ziegler et al. [Ziegler and Stiller, 2009] describe de generation of candidate piecewise trajectories considering position, velocity, acceleration and time (seven dimensions). In McNaughton et al. [McNaughton et al., 2011] a similar technique is implemented, arguing that the algorithm has a high computational cost and therefore implementing a graphic processing unit (GPU) to reduce this computation time. Moreover, in both approaches, the search space has to be pruned in order to arrive to a solution in time, introducing sub-optimality. On the other hand, Rapidly exploring Random Tree (RRT) has also been implemented to plan simultaneously a path and a speed profile. In [Jeon et al., 2013] RRT* is implemented as an on-line method, considering limit accelerations for the vehicle. This provides a solution that is somehow continuous but not

smooth, making it jerky and uncomfortable.

Having a smooth geometric path for an automated vehicle to follow is the first step to have a comfortable trajectory. The smoother the path, the gentler the lateral accelerations coming from speed variations. Comfort relates in an inversely proportional way to acceleration and Jerk; the higher the acceleration and its relative change (Jerk), the higher the discomfort [ISO, 1997]. EU interactIVe project evaluated lateral and longitudinal accelerations related to human drivers [Bosetti et al., 2014]. Statistical summary data shows that humans prefer to separate lateral and longitudinal actions, when the action involves important acceleration rates. However, data showed constant joint actions (lateral and longitudinal) under $3m/s^2$. Moreover, the study is coherent with [ISO, 1997], used in the present doctoral work to stablish comfort constraints.

The clear next step is to plan a speed profile that leads the vehicle from point A to point B. The speed planner has to be time effective, with an optimal comfort, complying speed and acceleration limits. The comfort is evaluated as the minimization and smoothness of acceleration and jerk profiles, while maintaining a coherent speed profile with respect to traffic rules, the geometry of the path and the lateral accelerations associated to it.

Speed profile planning is addressed in the literature with the Jerk Limitation method implemented in [Liu, 2002], [Haschke et al., 2008] and [Villagra et al., 2012]. In this method, the jerk is set as three different levels, according to whether the vehicle needs to accelerate, cruise or decelerate (details are explained in Section 3.4.3a). However, in this case the jerk is discontinuous and the acceleration profile is linear. Moreover, the speed is computed in terms of time, having inaccuracies in the distance covered over velocity changes due to delays in the control systems, deficiencies of the system adding noise, actuators' time reaction, among others.

To overcome these issues, a new speed planning algorithm implementing quintic Bézier curves is presented. These curves are continuous and smooth, meaning that the jerk profile is also continuous and smooth. The proposed method computes the velocity in terms of the length of the path, instead of time, reducing the inaccuracies described before. Contribution of the present section are described as follows:

- Compute a smooth and continuous speed profile considering acceleration limits (longitudinal, lateral and total acceleration) according to ISO 2631-1 standard [ISO, 1997].
- Address the distance error problem by associating the speed profile in the path speed planner instead of the time.
- Compare the Jerk Limitation implementation and our proposal with quintic Bézier curves.

To achieve the latter, the Jerk Limitation algorithm has also been implemented in equal conditions as the Bézier algorithm to have a proper comparison. The implementation of both algorithms is presented in the following.

3.4.3a Jerk limitation speed planner

The computation of a comfortable speed trajectory with the Jerk Limitation method, consists on constructing a proper jerk function $j(t)$. It provides speed and acceleration profiles without abrupt changes [Liu, 2002]. Figure 3.8 depicts in green the seven regions of jerk action for a speed profile.

$$j(t) = \begin{cases} J_{max}, & \text{if } t = I, VII \\ 0, & \text{if } t = II, IV, VI \\ -J_{max}, & \text{if } t = III, V \end{cases} \tag{3.10}$$

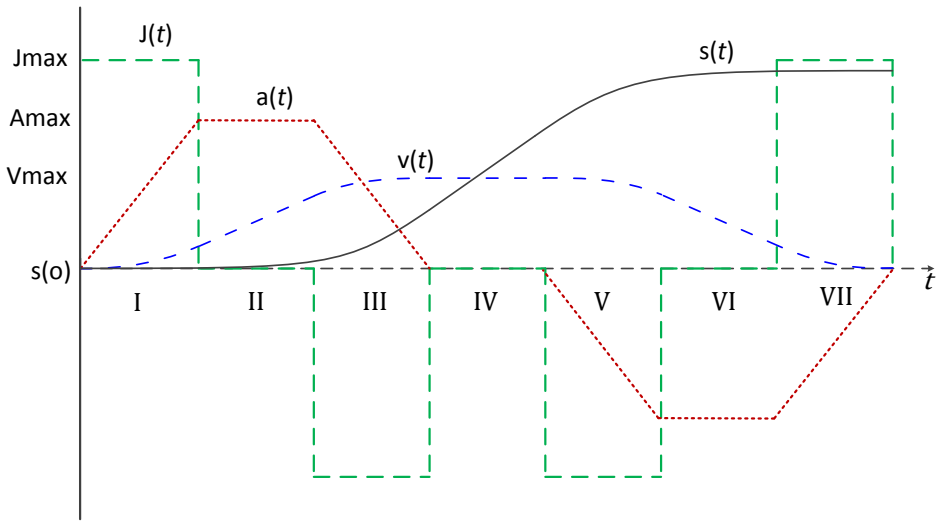


Figure 3.8: Intervals for the speed profile generation

Equation 3.10 shows the jerk assignment according to the time intervals t, from I to VII (as depicted in Figure 3.8). $Jmax$ is chosen as a small value to ensure comfort and smooth changes in acceleration. In [Villagra et al., 2012], this constant value is chosen as $1m/s^3$. The duration of each time interval will depend on the acceleration limit and the distance available to accelerate or decelerate. Based on the given Jerk, the acceleration, speed and distance profiles are computed in Equations 3.11, 3.12 and 3.13 respectively.

$$a(t) = a_0 + j_0 t \tag{3.11}$$

$$v(t) = v_0 + a_0 t + \frac{j_0 t^2}{2} \tag{3.12}$$

$$s(t) = s_0 + v_0 t + \frac{a_0 t^2}{2} + \frac{j_0 t^3}{6} \tag{3.13}$$

where t is the current time interval; and j_0, a_0, v_0, s_0 are the initial jerk, acceleration, speed and distance respectively for the current interval t. The duration of each time interval can be computed as:

$$t_I = t_{III} = t_V = t_{VII} = \frac{A_{max}}{J_{max}} \tag{3.14}$$

$$t_{II} = t_{VI} = \frac{V_{max} - V_0}{A_{max}} - \frac{A_{max}}{J_{max}} \tag{3.15}$$

where A_{max} and V_{max} are the maximum values for the acceleration and the velocity respectively. To calculate the cruise time of the vehicle (interval IV of Figure3.8), it suffices to compute the distance needed to accelerate (intervals I, II, III) and decelerate (intervals V, VI, VII), in relation with the whole path available, as follows:

$$S_{(t_I - t_{III})} = (V_{max} + V_0)\frac{A_{max}}{J_{max}} + \frac{\left(V_{max} - \frac{A_{max}^2}{2J_{max}}\right)^2 - \left(V_0 + \frac{A_{max}^2}{2J_{max}}\right)^2}{2A_{max}} \tag{3.16}$$

$$t_{IV} = \frac{D_{total} - S_{(t_I - t_{III})} - S_{(t_V - t_{VII})}}{V_{max}} \tag{3.17}$$

where Equation 3.16 is the distance needed for the acceleration and deceleration segments, however this last is with respect to V_{end} instead of V_0. Equation 3.17 gives the cruise time of the vehicle at V_{max}. Here D_{total} is the path distance, $S_{(t_I - t_{III})}$ is the distance to accelerate and $S_{(t_V - t_{VII})}$ is the distance to decelerate. For further details in the equations, the reader is referred to [Liu, 2002], [Villagra et al., 2012]. The limit velocity in all equations (V_{max}) is given by the speed limit of the road. The acceleration limit—lateral and longitudinal (A_{max})—is set by the user as the desired comfort level with respect to [ISO, 1997]. Levels are: *Not uncomfortable* ($0.315m/s^2$), *A little uncomfortable* ($0.63m/s^2$), *Fairly uncomfortable* ($1.0m/s^2$), *Uncomfortable* ($1.6m/s^2$) and *Very uncomfortable* ($2.5m/s^2$).

To compute the lateral acceleration limits according to [ISO, 1997], the maximum speed is set for each point in the pre-planned geometrical path, in such a way that the lateral acceleration with respect to the curve is not greater than the limits (see Equation 3.18).

$$V_{max} = \sqrt{\frac{a_w}{kC_{max}}} \tag{3.18}$$

$$a_w = \sqrt{{k_x}^2 {a_{wx}}^2 + {k_y}^2 {a_{wy}}^2 + {k_z}^2 {a_{wz}}^2} \tag{3.19}$$

where, $k_x, k_y = k = 1.4$, $k_z = 1$ are multiplying constants from [ISO, 1997]. a_w is the Euclidean sum of a_{wx}, a_{wy} and a_{wz} which are the root mean square accelerations in the different axis. Since the vehicle operates in a two dimensional space, the a_{wz} component of Equation 3.19 is neglected. C_{max} is the maximum curvature of a turn (max point when evaluating Equation 3.7 through t).

3.4.3b Quintic bézier speed planner

Bézier curves generation has been widely applied in the robotics field, however, normally applied to the physical trajectory rather than speed profiles. The novel speed planning approach benefits from their low computational cost, invariability, smooth and continuous characteristics to construct a comfortable speed profile for passengers in automated vehicles. To build the Bézier curve it suffices to set the control points of the curve (P_0 to P_n) in the desired shape of the curve. Equation 3.1 shows the definition of a Bézier curve of nth degree, where t parametrizes the curve from zero to one. In the following, the considerations to conform a speed profile with this kind of curves are explained.

Computation of Bézier curves for a smooth profile: The implementation of Bézier curves in speed planning requires joints between curves to be continuous. Moreover, jerk continuity is only achieved if the speed profile has G^2 continuous joints between curves. To achieve this quintic Béziers are implemented, since is possible not only to joint two consecutive curves, but also curves and straight segments (e.g. accelerate and then cruise speed), without losing continuity.

Figure 3.9 depicts a quintic Bézier curve. Control points are set in a symmetrical way, with distance d between them. This setup allows an intuitive manipulation of the curve, where the maximum slope is at $t = 0.5$, the change in the speed is represented by V; and D is the distance needed for the speed change. For optimal longitudinal speed generation without violating the maximum acceleration of Equation 3.19 and, at the same time, minimizing the jerk, is necessary to optimize the curve. Firstly, acceleration limits are set as follows:

$$M(t) = \frac{\partial V}{\partial D} \tag{3.20}$$

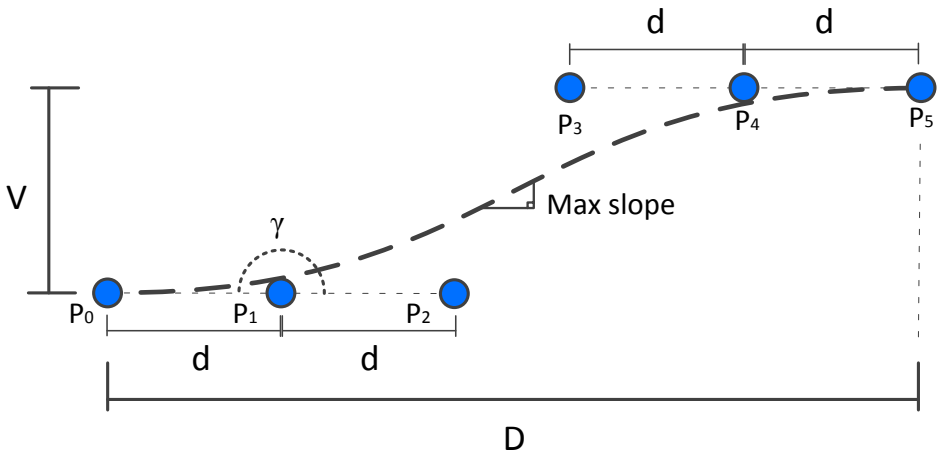


Figure 3.9: Fifth degree Bézier curve and its control points

$$A_{max} \geq M(t)V(t) \tag{3.21}$$

$$M(t) = \frac{\mathbf{B}'_y(t)}{\mathbf{B}'_x(t)} \tag{3.22}$$

where $M(t)$ is the slope of the Bézier curve, ∂V is the differential velocity change over the ordinate axis, ∂D is the differential change over the abscisa axis and $V(t)$ is the velocity value at t (keep in mind that t represents the parameter of Equation 3.1). $\mathbf{B}'_y(t)$ and $\mathbf{B}'_x(t)$ are the Cartesian derivatives of the Bézier curve. The inequation presented in Equation 3.21 develops from algebraic manipulation after including the temporal relationship between distance and velocity in Equation 3.20 (i.e. multiply and divide by the differential time step). Equation 3.21 is presented in this form due to the abscissa axis being distance instead of time. The optimization of the curve is done iteratively, looking for the lowest curvature profile (smoothest jerk).

$$k(0) = \frac{4\sin(\pi - \gamma)}{5d} \tag{3.23}$$

$$k(t) = \frac{300Vd(2t^3 - 3t^2 + t)}{(5\sqrt{((D-5d)(6t^4 - 12t^3 + 6t^2) + d)^2 + V^2(6t^4 - 12t^3 + 6t^2)^2})^3} \quad d > 0 \tag{3.24}$$

$$d = \frac{1.875(V - DM)}{5M(1 - 1.875)} \tag{3.25}$$

From evaluating Equation 3.7 at $t = 0$ and some algebraic manipulation, Equation 3.23 gives the curvature at the beginning of the curve ($t = 0$) and serves as proof that setting γ to π (i.e. the angle formed between $\overrightarrow{P_0P_1}$ and $\overrightarrow{P_1P_2}$ equal to π) is sufficient to ensure zero curvature at the beginning of the curve. Thus, to achieve G^2 continuity with previous segments it suffices to set P_0, P_1 and P_2 in the same horizontal vector, as in Figure 3.9. A similar process can be found for the other extreme of the curve (i.e. $t = 1$). Equation 3.24 presents the curvature for this specific

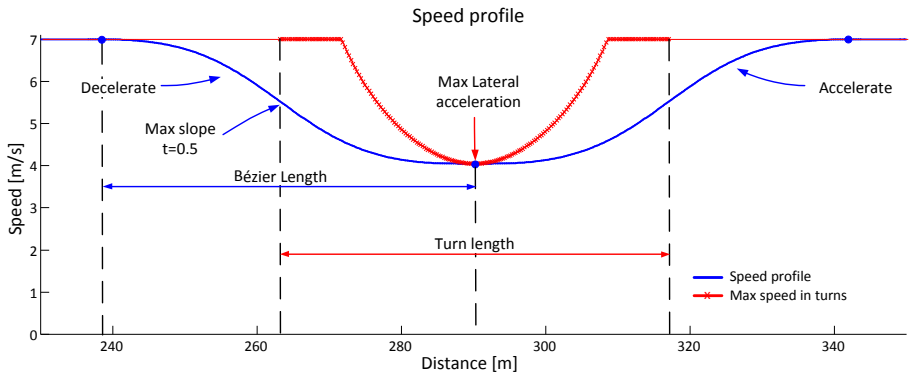


Figure 3.10: Speed profile presented in blue. Maximum speed in red (limited to $1m/s^2$ of acceleration)

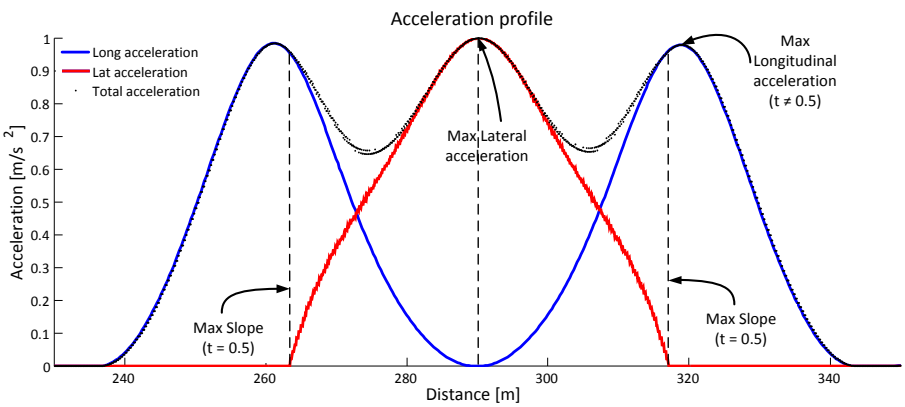


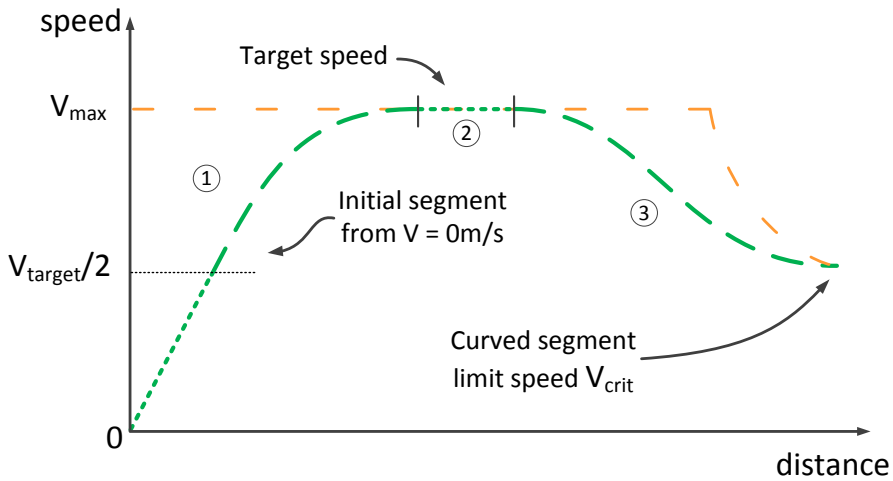
Figure 3.11: Absolute longitudinal and lateral acceleration profiles in blue and red respectively. The euclidean sum is represented in black

setup of the Bézier curve[1], when curvature at the beginning and at the end are forced to zero. This assures that every change in the velocity will have zero curvature (and therefore zero jerk) at $t = 0$ and also at $t = 1$.

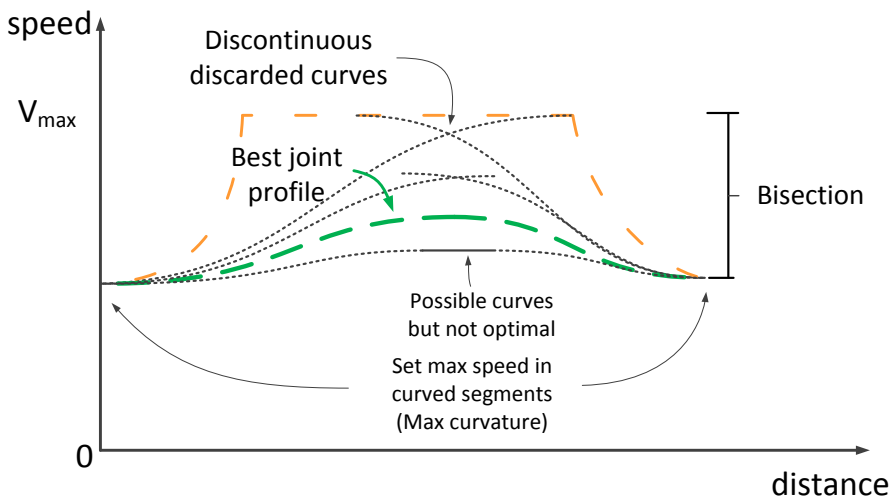
This configuration achieves G^2 continuity between joined curves. The continuity is only G^2 and not C^2 to avoid restrictive joints conditions. This allows to keep velocity, acceleration and jerk continuity, while preserving the flexibility of the curve. The expression in 3.25 comes from the development of Equation 3.22 at $t = 0.5$, which sets the control points inter-distance d for a given speed change V, critical slope M and distance D.

Considerations on acceleration limits: Comfort limits in ISO 2631-1 [ISO, 1997] describe acceleration actuating in the whole body. In a vehicle, longitudinal and lateral accelerations are the ones taken into account since the vehicle navigates in a plane. To satisfy comfort limits in the plane, Bézier curves are only set at the beginning and at the end of the turning area for the speed profile. This is done aiming to accelerate as soon as possible and decelerate as late as possible, while satisfying comfort limits.

[1]The position of the control points will be defined by d. It has to be a positive number different from zero to avoid cusps (infinite curvature profiles)



(a) Initial speed generation



(b) Selection of the best target speed to avoid discontinuities

Figure 3.12: Speed planning process: Choosing the best configuration.

The placing of acceleration and deceleration curves depend on the curvature profile associated to physical paths. Figure 3.10 presents the placing of Bézier curves while computing the speed profile. It is shown in blue and speed limits from Equation 3.18 are shown in red. Each curved stretch in the path is evaluated to find three key points: 1) Beginning of the curve; 2)Maximum curvature point; and 3)End of the curved segment. Bézier curves as applied in the present approach provide maximum longitudinal acceleration around $t = 0.5$ or near the middle of the curve. From this, it is possible to set speed changes in a way that it decouples the maximum longitudinal from lateral accelerations. It suffices to set the middle of the speed curve to the beginning point of the curved segment in the path (for situations when half the speed curve distance D is less than the distance to reach the maximum curvature in the turn). This is shown in Figure 3.10 at $265m$. Otherwise, both curves (deceleration and accelerarion profiles) will be joined at the maximum curvature point ($290m$ in Figure 3.10).

Acceleration profiles of the previous setting are shown in Figure 3.11. Longitudinal and lateral acceleration profiles are presented in blue and red respectively. Blue curves which represent

absolute longitudinal acceleration arrive to their limit (i.e. $1m/s^2$ for this example) just before encountering the beginning of the physical curve (lateral acceleration depicted in red). This is done to force one component to zero while the other one is at is maximum (see Equation 3.19). The total acceleration is depicted in the same figure as a black line, showing that the maximum total acceleration of $1m/s^2$ is never surpassed. Having this in mind, the approach can be explained as an optimization problem:

- First, optimal Bézier curves are computed. Equation 3.21 is initially set at $t = 0.5$, taking the maximum slope as $M_{(0.5)} = A_{max}/V_{(0.5)}$. With this first slope value, the control points inter-distance d can be computed with Equation 3.25 and D is biased as two times the speed change ($D = 2V$). Since the objective is to have the minimal jerk (proportional to the curvature, i.e. Equation 3.24), the algorithm searches iteratively D seeded by a value of $2V$ to find the minimum curvature profile (e.g. bisection since the algorithm looks in the region of the global minima) assuring the minimal jerk for this critical slope value. Since the critical slope is not necessarily at $t = 0.5$ (normally is higher, see Figure 3.11) due to the characteristics of velocity manipulation with respect to distance (see Equation 3.21), a new iteration step is needed. By varying the slope M the speed profile that complies with limits imposed by Equation 3.21 is found. This constraints the longitudinal acceleration of the speed profile.

- The generation of these curves for acceleration and deceleration profiles is done simultaneously, being V_{target} the target velocity for the current stretch (in a first step, it will always be V_{max}, see Figure 3.12a). V_{crit} is the velocity limit in the highest curvature point of turn segments in the geometric path. Three main stages are depicted in Figure 3.12a to create the speed profile, considering the correct placing of Béziers to decouple longitudinal and lateral accelerations. These allow to comply with lateral accelerations throughout the path (see Equation 3.18). Figure 3.12a shows the beginning of a speed profile from a stop point or the initial position of the vehicle, up to the first curved stretch. The acceleration curve aims to take the speed of the vehicle to the maximum allowed, following previous steps to compute the Bézier curve. To avoid slow starts, the lower part of the curve is replaced with a straight line that still ensures the G^2 continuity (see Section 1 in Fig 3.12a). The deceleration curve is generated taking the speed of the vehicle from V_{target} to zero (if the vehicle needs to stop completely) or to a limit speed constrained by lateral acceleration considerations in curved segments as in Equation 3.18 (see Section three in Fig 3.12a). Section two in Figure 3.12a only exists when there are no discontinuities between the acceleration and deceleration profiles and V_{max} has been reached.

- When the generation of optimal accelerating and decelerating Bézier curves creates discontinuities (see dotted black lines in Figure 3.12b), the algorithm looks for the best concatenation speed by varying V_{target}. Bisection is implemented, keeping continuous segments and discarding discontinuities. Then the best of the continuous approach is chosen (see green line in Figure 3.12b).

Once the speed profile is generated, the algorithm proceeds to merge it over the physical path to create the future trajectory for the vehicle. The trajectory is only planned up to a horizon, allowing the planner to create the local trajectory in real-time.

An example for the speed profile generation with quintic Béziers is presented in Algorithm 2. Lines one to three get the local path information. Line four sets the first approach for the speed profile, as in Figure 3.12a. If this first approach is discontinuous, lines six to 24 search for a proper

Algorithm 2 Speed profile computation

```
Start
Read algorithm properties
Get geometrical data of the local path—distance to points in the path, curvature, maximum
speed, etc.
Set acceleration and deceleration curves
if The speed profile is discontinuous then Find a continuous set
    while Best continuous speed profile not found do
        Get new V_target                                          ▷ Bisection applied
        Set new acceleration and deceleration curves
        Evaluate continuity
        if Curves are discontinuous then
            if V_target reached search limits then Stop the search
                Break the while loop
            else
                Continue the search
            end if
        else Curves are continuous
            if The best continuous set of curves has been found then
                Break the while loop
            else
                Continue to next iteration
            end if
        end if
    end while
end if
if A curve set has been found then Provide the trajectory
    Merge the new speed profile with the planned path
else No curve was found
    Inform control stage for safety actions
    Inform decision stage requesting new path strategy
end if
Send the speed profile to the control stage
End
```

V_{target} speed that gives a continuous curve set implementing a bisection approach. Then if a curve set is found, the planner merges the speed with the path and sends the speed profile to the control stage (lines 25-26 and 31). If no answer is possible, the algorithm notifies control and decision stages to execute safety actions and to provide a new strategy, respectively (lines 27-29).

3.4.3c Speed planners comparison

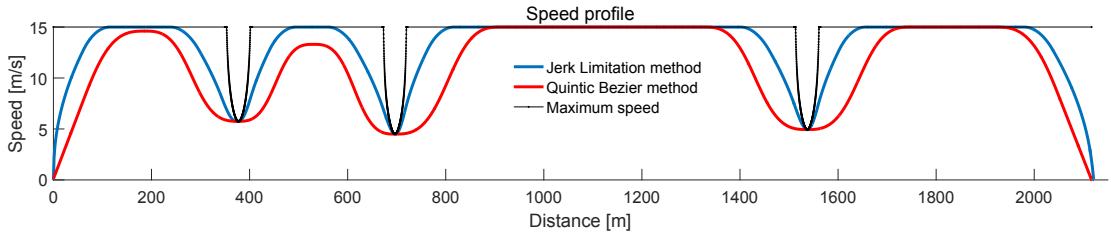
Two speed profile generators have been described so far, i.e. quintic Béziers method and jerk limitation method. Both planners are compared over a geometrical path with three turns that emulates an urban environment, implemented in Matlab. The vehicle needs to travel $2.1Km$ approximately, accelerating and decelerating avoiding to overpass the comfort acceleration (i.e. $1.6m/s^2$ for this experiment) but attempting to achieve the maximum speed possible at all times (the maximum speed is $15m/s$ for this urban scenario).

Figure 3.13a presents the speed profile for the path that encounters turns at $375m$, $694m$ and $1535m$, being consecutive the two first ones, where the second one is curvier than the first one. The black line represents the maximum speed which is lower in turns, complying with lateral acceleration limits (see Equation3.18). The Jerk Limitation method is depicted as the blue profile, with a more aggressive behavior than the Quintic Bézier method (depicted in red). In Figure 3.13b the acceleration profile for the Quintic Béziers only surpasses the maximum acceleration when braking (4% overshoot at most), due to the simulated vehicle inertia and delays in the control stage (theoretical results show no overshoot, see Figure 3.11). Figure 3.13c depicts the acceleration profile associated to the Jerk Limitation algorithm. The acceleration is continuous but non-smooth. It is important to notice that, even if the longitudinal accelerations are within limits, the euclidean sum in Equation 3.19 overshoots at almost each turn (i.e. the total acceleration overshoots by a factor of at most 44%).

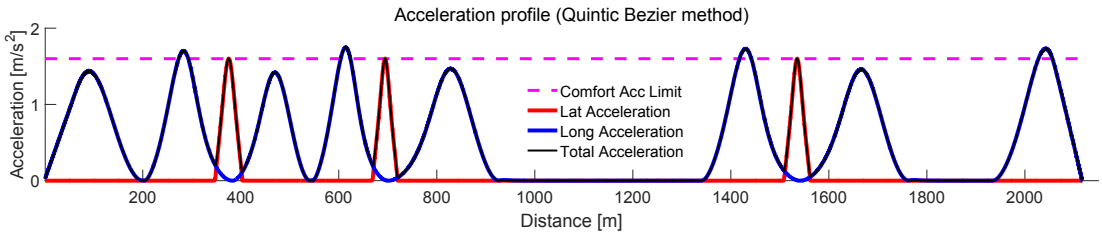
Notice that at the second turn of Figure 3.13c, the lateral acceleration (in red) surpasses the limit (marked by the dotted line). This is due to: 1) The speed profile surpasses the maximum velocity allowed due to distance errors inserted by the time-based planner, the delays in control and the vehicle model; and 2) The planner does not have the capacity to separate deceleration and acceleration curves as the Quintic Bézier method and therefore, the lateral acceleration (in red) goes out of bounds.

Figure 3.13d presents the jerk response for the Jerk Limitation method in blue and the jerk profile for the proposed method (quintic Béziers) in red. Notice that the blue line is not continuous, neither smooth. On the other hand, the proposed method (quintic Béziers) assures a continuous and smooth jerk. Values of the proposed approach outperform the previous method and are below $0.3m/s^3$ most of the time. Figure 3.13e presents the behavior of the controller and vehicle model when following the quintic Bézier speed profile. The performance is good with approximately $1.2s$ of stabilization time with respect to the speed profile for the model and slightly less for the control signal (i.e. $0.7s$).

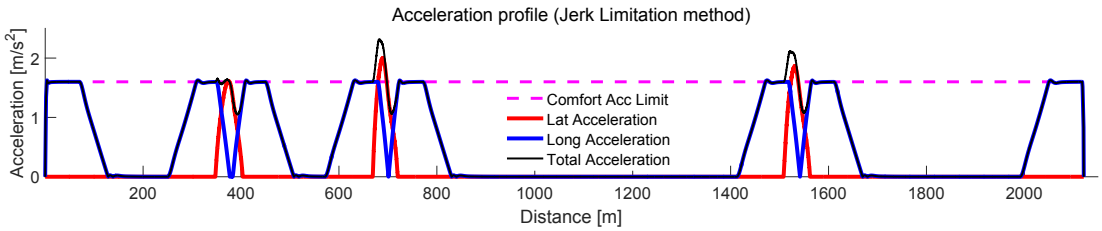
Up to this point, the theoretical framework of the proposed functional architecture has been presented. The generation of the trajectory is done in two separate steps. First, the geometric path that the vehicle is going to follow is generated (for turns and intersections), minimizing the curvature as much as possible, while keeping G^1 continuity. Then, the speed profile is generated with respect to the curvature profile of the path and comfort acceleration limits of the longitudinal, lateral and total accelerations felt by the passengers. The proposed motion planning architecture can address most of the cases presented in urban environments with this configuration. However, special scenarios like roundabouts present a specific challenge that needs to be addressed differently. The proposed solution in roundabouts is presented in the following section.



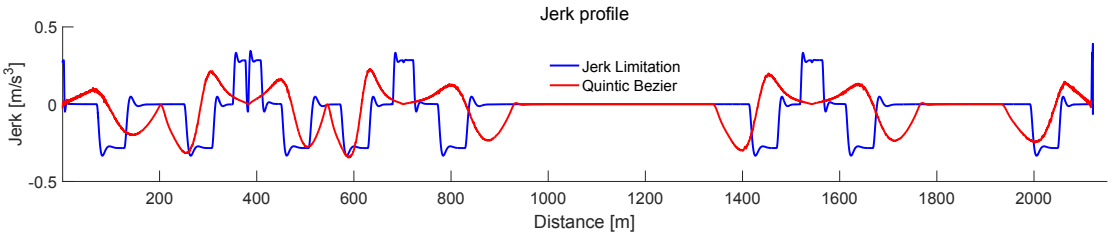
(a) Speed profile of both methods



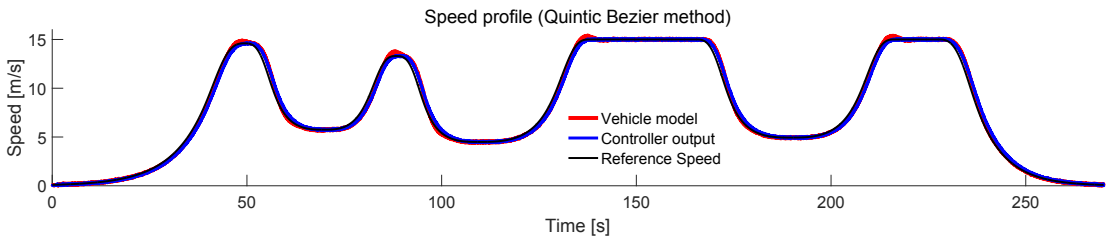
(b) Acceleration profile from the Quintic Bézier method



(c) Acceleration profile from the Jerk Limitation method



(d) Jerk profile comparison of both methods



(e) Controller and vehicle model behavior

Figure 3.13: Comparison between the jerk limitation method and the proposed quintic Bézier method.

3.5 Roundabouts

In the last two decades, roundabouts have been increasingly considered as a good alternative to address intersections, especially in urban areas [Manage et al., 2003]. They provide a number of safety, operational, and other benefits when compared to other types of intersections [Tracza and Chodura, 2012]. A detailed definition of the benefits of roundabouts in urban environment–i.e. compact size, operational efficiency, traffic safety and calming, access management, aesthetics and environmental benefits, are presented in [Rodegerdts et al., 2010] and [Rice, 2010a].

Roundabouts are relatively recent on urban environments, making difficult for some drivers—e.g. elderly people—to know how to deal with them[2]. Figure 3.14a shows the proper circulation in function of the exit according to the French road circulation code at roundabouts [3,4]. DARPA Challenges in the United States from 2004 to 2007 are among the most important automated vehicle competitions. Although roundabouts were out of the defined scenarios, several driving functions were shown, including traversing intersections [Ferguson et al., 2008].

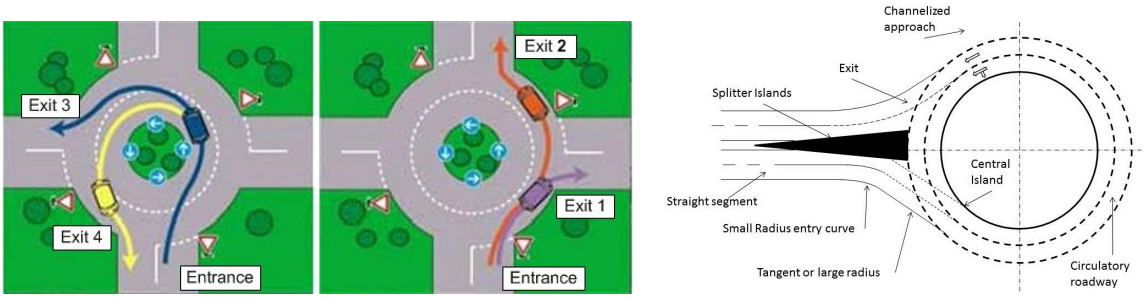
From the safety point of view, roundabouts have demonstrated positive effects worldwide. In United States, Robinson et al., [Robinson et al., 2000] concluded that they reduce the crash problems at intersections, approximately by 30%, compared with signalized intersections. In Europe, De Brabander and Vereeck [De Brabander and Vereeck, 2007] showed some conclusions on a Flanders' traffic study between 1994 and 2000, indicating that roundabouts can reduce by 39% the number of accidents. On the negative side, the driving rules at roundabouts are unknown for some drivers, causing an increment of 28% of vulnerable road user injuries compared with classical signalized junctions [De Brabander and Vereeck, 2007]. This fact turns roundabouts into suitable scenarios for automated driving, maintaining traffic flow benefits and removing safety concerns.

From the traffic flow point of view, there are several studies in the United States and China that show how some drivers do not have enough skills experiencing in roundabouts, generating traffic jams in the vicinities [Abaza and Hussein, 2009, Bai et al., 2009]. Roundabouts with and without traffic-signal controls have been intensively studied and widely used [Xiaoguang et al., 2004]. Despite of these efforts, roundabout applications have been only considered from the management point of view, providing reference speed control [Bai et al., 2009, Molinete et al., 2009, Yu and Qin, 2009].

From a classification point of view, a roundabout-dedicated Federal Highway Administration (FHWA) Technical Summary [Rodegerdts et al., 2010] divided them according to the size and number of lanes having: 1) Mini-roundabouts–a type of intersection that can be used at physically-constrained locations in place of stop-controlled or signalized intersections to help improve safety problems and reduce excessive delays at minor approaches [Robinson et al., 2000]; 2) Single-lane; and 3) Multi-lane roundabout.

Path and trajectory planning research has been very active on the intelligent vehicles field in last years (as stated in Chapter 2, see also [González et al., 2016b, Paden et al., 2016]). However, trajectory generation in roundabouts has not been further investigated, specially in real environments. Two main demonstrations have effectively tackled roundabouts: 1) Broggi et al. [Broggi et al., 2014] presented the PROUD test, involving different driving scenarios as highways, rural roads and urban roads (involving six roundabouts). The planning stage was composed of a database of precomputed clothoids, from which the best path was selected, targeting the center of the lane. The road layout was detected implementing vision algorithms, as well as obstacles and other road

[2] http://www.roundaboutsofbritain.com/
[3] http://www.securite-routiere.gouv.fr/
[4] http://www.mt.public.lu/transports/circulation/code/



(a) Roundabout behavior taking different exits.

(b) Definitions based on roundabout design in [Rice, 2010b].

Figure 3.14: Roundabout behavior and definition.

users; and 2) Ziegler et al. [Ziegler et al., 2014a] developed a planner that was based on global waypoints passed through a numerical optimization approach to plan its way through urban environments and perform the Bertha-Benz memorial route, managing several roundabouts. Stereo cameras detected the road layout, the obstacles and other road users.

This section presents a general approach using geometrical roundabout definitions in a digital map (center in 2-dimensional Cartesian coordinates, the radius, the number of lanes and the entrances and exits) to build a real-time reference path using parametric equations. The path is then given to the control stage where a lateral controller is able to follow the geometrical reference. Although, different control architectures for automated vehicles have been proposed in the literature, most of them consider perception, decision and control stages as the key ones (see Chapter 2). The present approach can be easily integrated in the decision and path planning modules, because its modularity permits an adaptability of the algorithm to any automated vehicle architecture. The main contributions of this section are:

- G^1 continuous roundabout entrance and exit by using an improved generation of the parametric control points.
- Path continuity, curvature smoothness and comfort constraints considerations for in-roundabout driving.
- Real time path generation considering two-lane roundabouts, including lane-change and taking different exits.

3.5.1 Roundabouts definition

A roundabout can be described as a composition and convergence of several intersections. According to the specific driving area, some parameters may vary as: number of entrances/exits, lanes, speed limit, geometric design or path alignment. A detailed review about the roundabout types is found in [Rice, 2010b]. The proposed approach faces the roundabout from the planning and control point of view, identifying three stages: 1) Entrance; 2) Circulatory roadway; and 3) Exit (see Figure 3.14b), as follows:

- **Entrance**: where the entry curve is generated. It covers the route from the straight stretch up to the beginning of the circular area. It can be divided in two sub-stages: the small radius entry and the tangent or large radius (see Figure 3.14b).

- **Circulatory roadway**: is defined as the middle stage, where generally circular shapes are used [Rice, 2010b]. It can be composed by one or multiple lanes, forcing lane-change maneuvers when entering/exiting the inner lane.
- **Exit**: is the final stage and it allows the vehicle to leave the roundabout. This stage is complementary to the entrance, having the same sub-stages: first the tangent or large radius and then the small radius exit.

Other parameters can significantly affect the automated roundabout path generation. Some of them are: 1) The central island, i.e. the raised zone defined in the center of the roundabout. It defines the minimum turning radius in the circulatory roadway; 2) Splitter islands, used in the entrance and exit of roundabouts, forcing the reduction of the entry speed; and 3) Channelized approach, describing dedicated exits in the roundabout to improve traffic flow. Other traffic rules (e.g. U.K.), are easily tunable due to the modular characteristics of parametric curves used in the approach (i.e. circle and Bézier equations).

3.5.2 Roundabouts constraints

There are some constraints to the proposed method imposed by both, the physical limitations of the experimental platform and the infrastructure design. Below, the main characteristics of type of roundabouts are:

- The maximum turning radius in the roundabout—considering the inner lane—has to be equal or bigger to the minimum turning radius of the vehicle with $3m$ width per lane. For the experimental vehicle used in this work, the minimum internal turning radius is 7.0 meters.
- The circulatory roadway is formed by two lanes and the roundabout comprises four exits.
- Perpendicular and tangential roads to the large radius of the roundabout are used.
- The interaction with additional vehicles—either automated or not—is out of the scope of this Ph.D. work. The reader is referred to [Milanés et al., 2012b] and [Muffert et al., 2012] for possible approaches addressing intersection and roundabout interaction.

Based on Sections 3.5.1 and 3.5.2, the following details the proposed path planning approach for automated vehicles in roundabouts. This approach is included into the functional planning architecture for automated vehicles presented in the thesis.

3.5.3 Automated Roundabout Algorithm

In Section 3.4.2, a description of the curve process used for intersections is presented. This process uses Bézier curves to provide continuous paths for the vehicle to follow. The approach is now oriented to overcome roundabouts and considers several parameters as the vehicle dimensions, speed limit, comfort accelerations and characteristic of the infrastructure (i.e. small radius entry curve, tangent radius and central island). Joints between different segments in the path are at least G^1 (geometrically continuous).

To meet roundabouts special shape, the curve generation was divided in three stages: entrance, circulatory roadway and exit. Left part of Figure (see Figure 3.15) shows the flowchart of the proposed algorithm. The first and the last stages can be modeled as standard intersections, but considering the small radius entry curve and the tangent radius, based on parametric equations as in

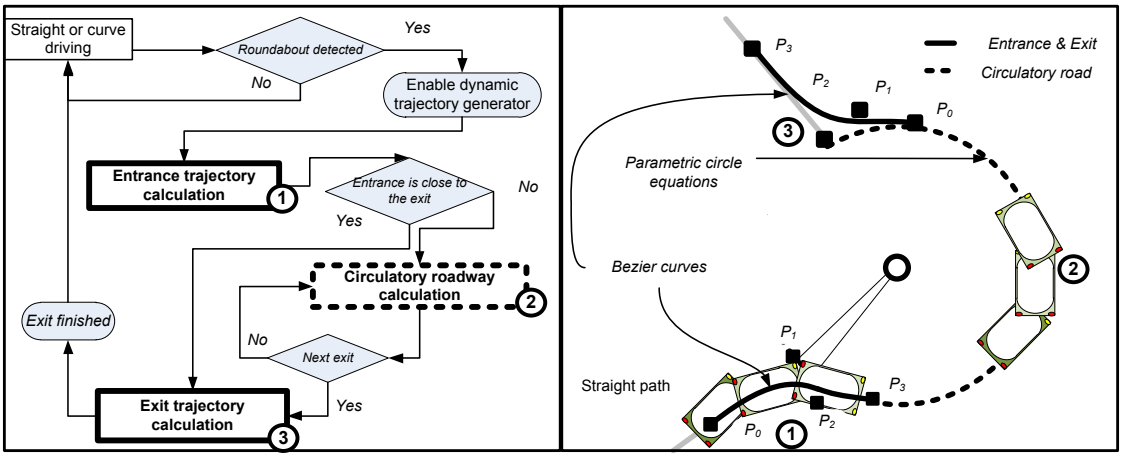


Figure 3.15: Roundabout phases and parametric path generation.

[González and Perez, 2013]. The polynomial is described as in Equation 3.1. Bézier curves of third and fourth degree are implemented according to geometrical characteristics for the entrance and the exit of the roundabout. The junction points require special attention to avoid discontinuities on the path. The circulatory roadway can be modeled as in [Pérez et al., 2011], using the parametric equations of the circle, considering also lane changes.

Right part of Figure 3.15 shows the three stages proposed for the roundabout. Once the entrance of the roundabout is detected, the first stage is generated. Its search space will be limited by half the distance between the entrance and the exit (around "2" in the right part of Figure 3.15). In cases where the entrance curve reaches half of the distance (meaning that the exit stage is close), the circulatory roadway is avoided and the exit stage is generated. On the other hand, if there is enough space for a circulatory roadway, it is generated using the circle parametric equations up to the exit stage.

Depending on the roundabout, different Bézier degree polynomials are generated. First, 3rd degree Béziers are tested according to an optimization function that takes into account the curvature of the generated Bezier curve, and the curvature error in the joining points of two curves. If no curve meet the requirements (not complying with road/vehicle constraints), 4th degree Béziers are implemented in a similar process to find the best curve. The extra control point coming from the curve degree elevation (P_2 point in Figure 3.16) permits to pull the curve closer to the splitter island, introducing more flexibility in the path and enabling configurations unreachable with 3rd degree Béziers. Figure 3.16 shows an example of the path generation with a 4th degree Bézier, where point P_2 is moved in-between the line generated from the curb to the splitter island (yellow squares), thus forcing the Bézier to stay within bounds (see the convex-hull in grey), while still pulling the curve towards the splitter island.

Figure 3.16 shows the characteristics of the path generation at roundabouts. In the entrance of the roundabout, the small radius entry curve, the tangent radius and the radius of the circulatory roadway are considered in the path generation.

In this sense, the position of the splitter island limits the Bézier convex-hull (gray region in Figure 3.16) which gives a secure internal band to the vehicle since the curve lies within the convex hull. Points P_3 and P_4 are set over the tangent line to the circulatory roadway to ensure soft curvature transitions in the path. The vector defined by $\overrightarrow{P_0P_1}$ describes the direction of the entry

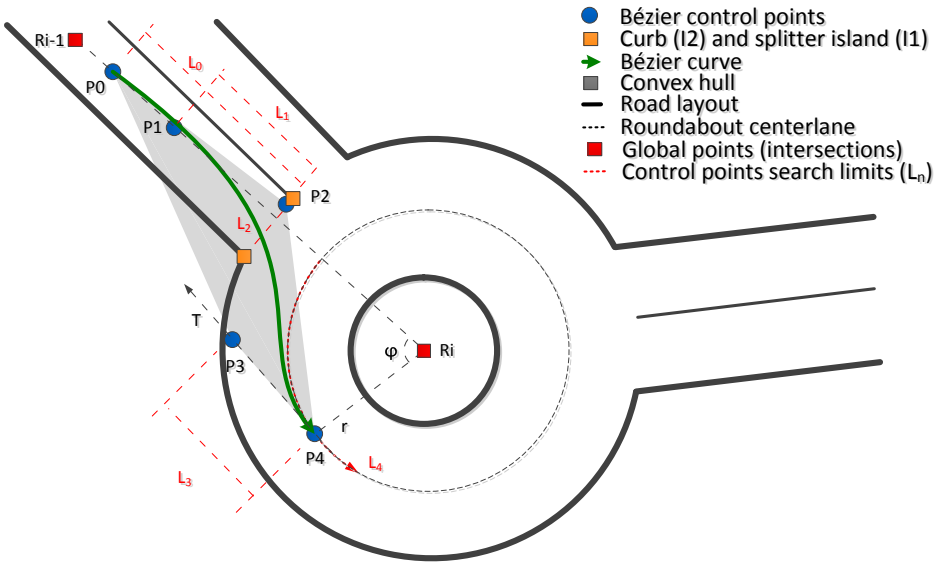


Figure 3.16: Solution for the roundabouts.

road in Figure 3.16, being $\overrightarrow{P_3P_4}$ the points defining the exit stretch when leaving the roundabout. The generation of the control points is as follow:

- The yellow points in Figure 3.16 describe the road layout (I_1 for the splitter island and I_2 for the curb), constraining the movement of point P_2 and ensuring the convex-hull does not surpasses the splitter island (the point lies within the line formed by these yellow points).
- The distance from the curve (green) to the curb and the splitter island (yellow squares) has to be at least half of the vehicle's with. If not, the curve is discarded (avoiding going over the curb).
- The maximum curvature in the Bézier curve, i.e. the maximum value from Equation 3.7, must be less than the maximum physical curvature of the vehicle (see Equation 3.8).

Figure 3.16 shows the area where the control points (from P_0 to P_4) are established. The position of the point P_0 is calculated as in Equation 3.26 and P_4 is represented by the Equation 3.30:

$$P_0 = R_i + L_0 \frac{R_{i-1} - R_i}{\|R_{i-1} - R_i\|} \tag{3.26}$$

$$P_1 = R_i + L_1 \frac{R_{i-1} - R_i}{\|R_{i-1} - R_i\|} \tag{3.27}$$

$$P_2 = I_1 + L_2 \frac{I_1 - I_2}{\|I_1 - I_2\|} \tag{3.28}$$

$$P_3 = P_4 + L_3 \overrightarrow{T} \tag{3.29}$$

$$P_{4_{(x,y)}} = \begin{pmatrix} r\cos(\arctan(\frac{R_{i-1y}-R_{iy}}{R_{1-1x}-R_{ix}}) + \varphi) + R_{ix} \\ r\sin(\arctan(\frac{R_{i-1y}-R_{iy}}{R_{i-1x}-R_{ix}}) + \varphi) + R_{iy} \end{pmatrix} \tag{3.30}$$

where L_0 and L_1 is the linear separation between P_0, P_1 respectively from R_i. R_i and R_{i-1} define the roundabout center and the previous intersection respectively. Please notice that L_0 is always greater than L_1 to avoid unexpected curve behaviors and both, L_0 and L_1, are always greater than the external radius of the roundabout. L_2 describes the position of P_2 inside the line $\overrightarrow{I_1I_2}$. Moreover, P_3 is defined by L_3 which is the distance from P_4 to P_3 in the direction of vector $\overrightarrow{T}$, which is a vector tangent to the roundabout at P_4. Please also notice that if only 3rd degree curves are needed, then P_2 is removed. Moreover, φ is defined by Equation 3.31:

$$\varphi = \frac{L_n}{r} \tag{3.31}$$

where L_n is the segment of arch formed between $\overrightarrow{R_{i-1}R_i}$ and P_4, and r is the radius of the roundabout (Circulatory roadway). The exit of the roundabout is managed as an entrance, using symmetrical conditions. The circulatory roadway segment of the roundabout is generated by parametric circle equations as follows:

$$Rb_x = r\cos(\varphi) + R_{i_x} \tag{3.32}$$

$$Rb_y = r\sin(\varphi) + R_{i_y} \tag{3.33}$$

where $Rb_{x,y}$ describes the path points generation inside the circulatory roadway of the roundabout. Each point is equally spaced, thus L_n changes at a rate of $0.5m$ until the circulatory roadway joints the exit stage. $R_{i_x,y}$ are the x, y components of the roundabout center R_i.

Having this in mind, control points are constrained to linear search spaces (defined by L_0, L_1, L_2, L_3 and L_4). The curve generation is done iteratively, being the limits and resolution for each of the search spaces as follows:

- For the external control points P_0 and P_4, L_0 will be constraint at maximum $20m$ from the external radius and L_4 at $20m$.
- For P_1, L_1 will be constraint at maximum L_0 distance from the external radius (forcing P_1 to always be further into the intersection than P_0, avoiding undesired behaviors). For P_3, L_3 will be limited to L_n.
- If P_2 exists (4th degree Béziers), L_2 will be constraint to $\left|\left|\overrightarrow{I_1I_2}\right|\right|$.
- Outer control points related to L_0 and L_4 will have a resolution of $2m$ (ten iterations per control point). Inner control points length L_1 and L_3 will also be covered in ten iterations, having variable resolution (since they are limited by L_0 and L_4 respectively. Finally, L_2 is covered in five iterations, presenting also a variable resolution depending on $\left|\left|\overrightarrow{I_1I_2}\right|\right|$.

Algorithm 3 describes the curve generation at entry and exit stages, where k is the curvature of the evaluated point in the curve at a given iteration (i.e. Equation 3.7 at a given t) and k_{max} is the maximum curvature value. k_{start} and k_{end} are the curvature continuity errors at P_0 and P_n respectively, these are computed to assess the continuity of the curve with respect to adjacent

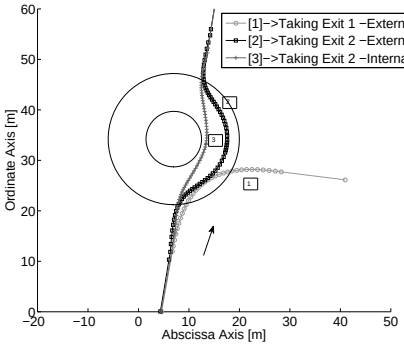
stretches, e.g. straight segments before the roundabout and curve segments inside the roundabout (see Figure 3.16). The parameter n is the Bézier degree. These data are stored in a dynamic array, rewarding when the value of k_{max} is lower than the vehicle's maximum feasible k. The best option is thus the one with the smallest reward value (minimizing the continuity error between segments).

Algorithm 3 Bézier generation in roundabouts

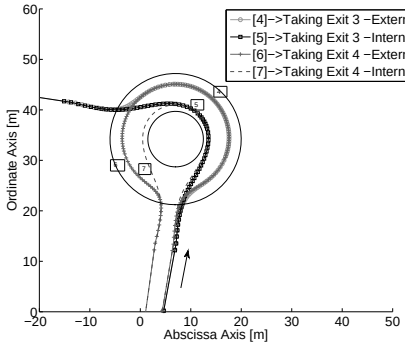
```
Start
Read algorithm properties
Get geometrical data of the route—radius, road width, etc.
for n=3 ; n≤4 ; n++ do
    Generate all control points
    while the evaluation process is not completed do
        Evaluate k_start, k_end and k_max from each curve
        if k_max is ≤ vehicle's maximum feasible k then
            Assign reward as max (k_start, k_end)
            Save control points and their associated reward
        else discard the curve
        end if
    end while
    if A curve has been found then
        Break the For loop and provide the optimal curve (minimum value for: max (kstart; kend))
    end if
end for
if No curve was found then
    Inform control stage for safety measurements
    Inform decision stage requesting new path strategy
end if
Generate the best curve (iterate through t)
End
```

Different entrances and exits of the roundabout have been evaluated, where the curvature, its derivative and the lateral and angular errors determine the circulatory roadway lane. Figure 3.17 shows the behavior of the proposed algorithm when taking different exits using both inner and outer lanes. For the scenario when the first exit is taken, the path only considers Bézier curves as in a standard intersection. This is because the exit is too close to the entrance, making the circulatory roadway negligible and applying the approach presented in Section 3.4.2. The other two paths depict the scenario when the second exit is taken. Despite the fact that the external lane is usually taken when going straight (exiting in the second exit), the use of the inner lane allows to illustrate the curvature evolution.

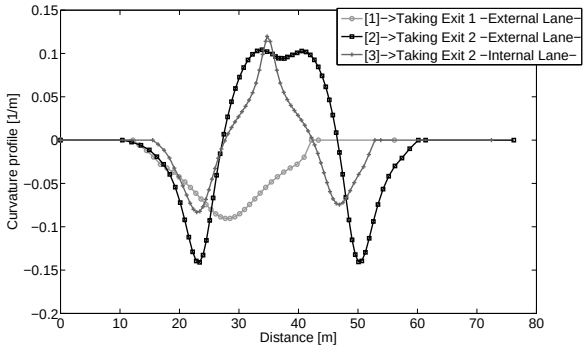
Figure 3.17c details the curvature in the three scenarios from Figure 3.17a. For the first case, the curvature is smoother and smaller than in the second and third cases. It is because the behavior is similar than a standard intersection and there are no inflection points in the curvature. However, the curvature is a little higher when taking the inner lane. Experiment two details the curvature generation taking the external lane. The derivative change has a better behavior on the experiment three (see Figure 3.17e). This shows how the comfort of the drivers is better in the third case, because the reduction of the lateral accelerations inside the roundabout.



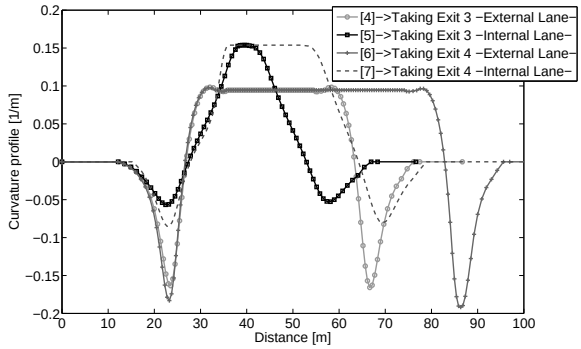
(a) Planning in the Roundabout -first and second exit-



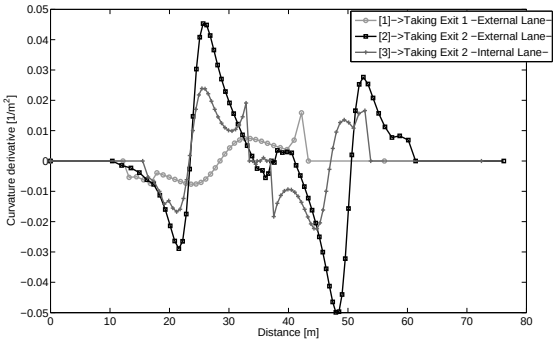
(b) Planning in the Roundabout -third and fourth exit-



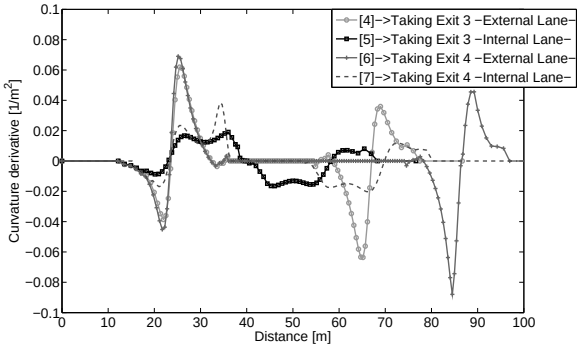
(c) Curvature -first and second exit-



(d) Curvature -third and fourth exit-



(e) Derivative of the curvature -first and second exit-



(f) Derivative of the curvature -third and fourth exit-

Figure 3.17: Evaluation of roundabouts with different radii, lane changes and exits

Path generation when taking the third and fourth exits following the French Road Circulation Code are plotted in Figures 3.17b,3.17d and 3.17f. Figure 3.17b depicts the Cartesian position of the trajectories generated. Curvature and its derivative are detailed in figure 3.17d and 3.17f respectively. At the beginning of Figure 3.17d, the entrance of the roundabout is depicted. When the outer lane is taken (see tests four and six in Figure 3.17b), the vehicle steers more than when taking the inner lane (i.e. tests five and seven in Figure 3.17b), causing a higher curvature in the entry and exit stages. In both cases, the use of the inner lane has a better behavior. The covered

distance is smaller (see Figure 3.17d) and the derivative of the curvatures have smoother changes during the whole maneuver (see Figure 3.17f).

3.6 Conclusions

In this chapter a novel real-time trajectory planning algorithm has been presented. The development and implementation of different algorithms for a safe and comfortable path and speed profiles is proposed for the generation of comfortable trajectories for automated vehicles to follow. The main contribution to the RITS automated vehicles architecture takes place in the decision stage with the proposed modular motion planning structure.

The singular points, set in the global planning module, describe the characteristics of the road. These are smoothed by the local planning module, implementing parametric curves, to increase comfort. Bézier curves are suitable to this task, since the positioning of its control points is intuitive and provide the best compromise between flexibility, continuity and low computational characteristics. The resulting geometrical path fits the vehicle kinematics/dynamics and the road layout constraints. The evaluation of the derivative of the Bézier curve at its limits (where $t = 0$ or $t = 1$) permits G^1 continuity between curves. Having a correct placing of the control points, adjacent singular points can have a G^1 continuous profile. The curvature is optimized throughout the path, as well as curvature joint discontinuities are minimized to reduce lateral accelerations.

At roundabouts, the proposed real-time algorithm also considers the road layout and the vehicle's constraints. It takes into account the roundabout radii, central point of the roundabout and the exit to be taken on the roundabout. The path-planning generation was split in three stages: entrance, circulatory roadway and exit. Each part has a specific path generation based on parametric curves. Different exits of the roundabout were considered based on the French Road Circulation Code showing a smooth curve generation, validating the proposed algorithm.

The proposed strategy is resolution complete, meaning that optimality depends on the discretized search of the control points position (if a solution exists in the current control points search resolution, the planner is able to find it). In case the curve does not exist, the planner notifies the control and decision stages (requesting safety measurements and a new strategy, respectively). However, the ability of introducing passengers' comfort in the planning process relies also over the speed profile generation in addition to the smoothness of the path. It is thus possible to create trajectories by fusing the previous smoothed path, with a dedicated speed profile, that complies with comfort constraints.

Having this in mind, a novel speed planner based on Quintic Bézier curves was also presented. The introduction of polynomial curves in the speed planning process significantly increase passengers' comfort by smoothing acceleration and jerk profiles. The generated speed profile is G^2 continuous, thanks to Bézier curves characteristics and a proper curve concatenation. Acceleration profiles have been evaluated, allowing to manage acceleration/deceleration profiles with good results. Longitudinal accuracy errors have been reduced, since the planner is distance-based, allowing a precise behavior throughout the geometric path.

The comparison between the proposed Quintic Bézier method and the Jerk Limitation method has been made to evidence a clear improvement in acceleration and jerk smoothness. This provides a more pleasant travel without exceeding comfort levels described in ISO 2631-1 standard [ISO, 1997]. The proof of concept showed the correct behavior of the vehicle in urban intersections, where the comfort acceleration is surpassed slightly, due to the response time of the vehicle's control stage.

Chapitre

Planification des trajectoires pour des environnements dynamiques

Below is a French summary of the following chapter "Trajectory planning in dynamic environments".

La méthode de planification de mouvement présentée dans le chapitre précédent fournit des solutions qui prennent en compte le confort du passager et les contraintes de la route (entre autres), lorsque l'environnement ne change pas. Elle exploite le fait que la configuration routière est connue a priori pour calculer la meilleure trajectoire possible sans tenir compte de la présence d'obstacles ou d'autres utilisateurs de la route.

La prise en compte des autres acteurs (par ex. : les piétons, les cyclistes et les véhicules) augmente considérablement la complexité de la conduite autonome lorsqu'il s'agit de gérer tous les scénarios. Il faut alors pouvoir prédire le comportement des autres et décider et calculer en temps réel la trajectoire du véhicule pour conserver un bon niveau de confort et éviter les collisions.

Le présent chapitre décrit l'algorithme proposé pour une planification de la trajectoire en milieu urbain en présence d'obstacles. Il étend l'approche introduite dans le chapitre précédent en ajoutant un élément en cascade appelé module d'évitement d'obstacle dans l'algorithme de planification locale. Celui-ci prend en compte les informations des algorithmes de perception et modifie la trajectoire initiale quand elle est en conflit avec un obstacle.

L'approche proposée se concentre sur la génération de trajectoires en temps réel pour les manœuvres d'évitement des obstacles. Les jonctions entre la portion du chemin d'origine (calculé dans l'environnement statique) et le nouveau chemin d'évitement sont continus. Cette approche est modulaire et agit uniquement sur la trajectoire déjà transmise au contrôleur (le module modifie la trajectoire déjà émise et ne modifie pas le comportement du planificateur de trajectoire original). Cet algorithme permet un calcul en temps réel et le retour à la trajectoire optimale précédemment calculée, une fois l'obstacle dépassé.

Chapter 4

Trajectory planning in dynamic environments

The development of motion planning algorithms presented in the previous chapter gives smooth solutions for comfort and road layout constraints (among others) when the environment does not change. Chapter 3 exploited the static characteristic of the navigable space (i.e. the road layout is known a priori) to compute the best possible trajectory without considering obstacles or other road users.

The inclusion of traffic agents (i.e. pedestrians, cyclist and vehicles) increases dramatically the autonomous driving complexity when it comes to handle all scenarios. Specifically, the main difficulties are: 1) more challenging situations arise from the unpredictable nature of the scenario, where the decision system needs to include the current and future predicted states of obstacles; and 2) higher reactivity is needed to deal with all situations, even unexpected ones, requiring a fast decisional process and real-time response to provide a collision free trajectory in time.

The present chapter describes the proposed solution for trajectory planning in urban environments when considering obstacles. It is based on parametric curves for a real-time generation of obstacles-free paths. The proposed solution for dynamic environments incorporates a cascade element in the local planning architecture (called obstacle avoidance module), which monitors the communication buffer between the trajectory issued by the planner and the control stage. It takes into account the information coming from the perception stage, supervising that the original trajectory is not in conflict with any obstacle in front of the vehicle.

The proposed approach focuses in real-time trajectory generation for obstacle avoidance maneuvers, where the front obstacle is always slower than the ego-vehicle. Joints between the original path (computed as described in Chapter 3) and the new avoidance path are G^2 continuous, implementing curve interpolation to achieve such continuity. The general design of the obstacle avoidance module permits to avoid static and dynamic obstacles, while remaining modular, acting only over the trajectory already issued to the controller (the module modifies the trajectory in the buffer and does not tampers with the behavior of the trajectory planner module). This allows a lower computation time and also returning to the optimal trajectory previously computed.

The present chapter is structured as follows: Section 4.1 describes the problem from the literature point of view and gives a first description of the proposed approach. Section 4.2 presents the proposed local planning architecture as a cascade system with developments in the previous chapter, where the main contribution is an obstacle avoidance module, capable of supervising the optimal strategy given by the trajectory planner and modifies the trajectory only if it is in conflict with an obstacle. This section is divided in two main blocks: 1) Real-time trajectory modifica-

tions, which gives the bases to assess future collision states and the generation of the new obstacle avoidance path; and 2) Obstacle avoidance/overtaking, which describes the implementation of overtaking maneuvers for obstacle avoidance in urban scenarios. Finally Section 4.3 presents the final conclusions of the chapter.

4.1 Problem description

Recent developments in the automated vehicle field are facing highways and urban environments. Highways, being the safest and less challenging environments, have already been tackled by different OEMs like Tesla, Mercedes, among others (see [Karush et al., 2016]). On the other hand, urban environments represent the most challenging scenario, because of its complexity and the number of road actors that interact with the vehicle. Different European projects have recently focused in urban driving like CityMobil, CATS, V-Charge, CityMobil2, deploying automated vehicle demonstrations across Europe, as well as individual actors such as Google, Uber and NuTonomy (these last two deploying automated taxi systems in Pittsburgh and Singapore respectively—but still keeping human drivers behind the wheel for safety reasons—). Moreover, reports from the E.U. [ERTRAC, 2015] and data analysis in the U.S. [Anderson et al., 2014] show that continuous research in automated vehicle technology (ranging from ADAS up to fully automated) can greatly reduce emissions, travel times and risks, while at the same time increase comfort, enable social acceptance of the technology and push forward traffic optimization. However, there is still a long way before full automation capabilities avoid all road accidents as shown in recent fatal crash reports from the E.U. [European Commision, 2015] and the U.S. [NHTSA, 2016], reporting difficulties in decreasing the number of fatal accidents in European and U.S. roads respectively.

When it comes to motion planning in urban environments, the dynamic characteristics of the navigation area and the diversity of road users represent the main difficulties. Previous efforts to cope with this problem as in [Hardy and Campbell, 2013], [Ziegler et al., 2014a], among others, presented approaches that optimized trajectories accounting for the ego vehicle path and possible conflicts with other obstacles, in present or future states. [Hardy and Campbell, 2013] focused on the collision probability to compute multiple spline paths at low speeds, allowing to handle intersections and passing static obstacles in simulated environments. On the other hand, real tests by [Ziegler et al., 2014a] implement a numerically optimized cost function, including velocity and accelerations changes, the road layout, static obstacles and the distance to the center of the lane. Dynamic obstacles are handled by adapting the vehicle speed rather than its path (which relies on global waypoints).

Given the broad aspect of motion planning—and the difficulty of having a general algorithm for all situations—the generation of a dynamic trajectory to avoid collisions is divided in two: intersection management and obstacle avoidance. Intersection management aims mostly at managing and organizing traffic within intersections (where paths are intrinsically conflicted), meaning that longitudinal planning and control is mostly applied to solve trajectory conflicts (see [Milanés et al., 2011] and [Zhang et al., 2015]). On the other hand, obstacle avoidance in automated vehicles relates to more complex lateral actions, where the common approach to avoid obstacles, follow road rules and have a human like behavior altogether, relies on overtaking maneuvers (i.e. lane changes and passing maneuvers, see [Naranjo et al., 2008], [Glaser et al., 2010] and [Milanés et al., 2012a]).

[Naranjo et al., 2008] describes lane change maneuvers for static and dynamic obstacle avoidance in urban environments relying on GPS and V2V communications. Two fuzzy logic controllers are implemented as a way to include human knowledge in the control strategy, where one controls

the lateral action when in straight line following and the other for the lane change maneuver. Distance for the first lane change is chosen with respect to the velocity of the vehicle (look-up table), assuming also that both lane changes (to avoid the obstacle and to return to the original road) will be of the same distance. Results presents effective obstacle avoidance maneuvers, where the vehicle reaches speeds of 30km/h and 55km/h for the static and dynamic obstacle avoidance test respectively. [Milanés et al., 2012a] also implemented fuzzy logic controllers for overtaking, relying on computer vision for obstacle localization and tracking. The approach adapted the lateral displacement with respect to the front obstacle width (motorcycle, car or truck), and also its speed with respect to the obstacles length (enabling also its implementation for pedestrian collision avoidance, as in [Llorca et al., 2011]). The avoidance maneuver is triggered when the distance reaches a comfort value from lateral acceleration. However, overall accelerations are not shown for evidencing the principal comfort parameters (lateral and longitudinal accelerations), as well as no trajectory is implemented and therefore the lateral behavior is biased, only to behave as the values of the fuzzy membership functions dictates, leaving no room for adaptability.

One of the first solutions to dynamic obstacle avoidance by an automated vehicle (from the trajectory planning perspective) is proposed in [Shamir, 2004]. Here, the overtaking maneuver is described as the optimal way to avoid obstacles in structured environments. It is a three phase maneuver composed of two lane changes and a passing phase. The approach focused in minimizing the overall kinetic energy during the maneuver, implementing curve interpolation with polynomials. The distance to overtake was found and tabulated by a numerical optimization algorithm (iteratively) with respect to velocity and acceleration constraints. However the approach stayed theoretical and no tests were conducted. [Keller et al., 2011] presented a path planning implementation for ADAS applications. Polynomial curves were implemented as a mean to compute the critical avoidance path, when braking is not the best solution (i.e. there is no time for a full stop or the driver is doing a contrary action over the pedals). Real tests showed good results for obstacle avoidance in straight roads.

Other approaches as in [Werling et al., 2010] and [Werling et al., 2011] point to planning strategies where the lateral and longitudinal actions are decoupled. Optimal control theory is implemented to optimize quintic polynomials that will form the possible trajectories from which the final trajectory is selected within a frame-rate of $100ms$. The approach allows to react to emergency and unexpected situations, however, the evaluation of lateral and longitudinal strategies is somewhat greedy and accelerations can increase quickly, neglecting comfort. A similar approach implementing spatio-temporal lattices have been used by [Ziegler and Stiller, 2009] and [McNaughton et al., 2011], where polynomial curves joint terminal manifolds in a lattice. The obstacle avoidance is done by evaluating each of the lateral and longitudinal strategies minimizing the collision risk with other vehicles at constant speeds. Specifically [McNaughton et al., 2011] generated trajectories "on the fly", implementing a GPU, whereas [Ziegler and Stiller, 2009] implemented an offline precomputed search graph. Both results where validated in simulation.

In the autonomous vehicle competition organized by the Hyunday-Kia group, [Chu et al., 2012] presented a local planning strategy that joined global waypoints with spline curves (interpolation). Then for static obstacle avoidance the path was "deformed" considering offsets in the curvature of the original spline path. This creates parallel path options to be followed by the vehicle and avoid static obstacles. The selection of the optimal path relied over safety and smoothness parameters weighted in a cost function. Stanford [Funke and Gerdes, 2016] also presented an interpolating approach for lane change incorporating friction constraints. This brings obstacle avoidance capabilities to their previous developments in motion planning [Funke et al., 2012], by generating suitable real-time clothoids for lateral emergency actions.

Curve interpolation for obstacle avoidance and lane change has also been used in the E.U. FP7 HAVEit project by [Glaser et al., 2010]. The highway-based ADAS implementation consisted on two phases: 1) trajectory generation minimizing collision risks; and 2) a more precise optimization with a cost function including comfort parameters, road layout, driving laws, among others, selecting the one with more resemblance to the driving mode chosen by the driver. [Petrov and Nashashibi, 2014] also implemented interpolation, modeling obstacle avoidance operations as three steps overtaking maneuvers: 1) Lane change; 2) Passing; and 3) Return to the right lane/original path (as in [Naranjo et al., 2008] and [Milanés et al., 2012a]). Polynomial curves were used to design lane change trajectories. However, both approaches were done only in simulated environments.

Having in mind the previous review of existing systems focused in obstacle avoidance (and the more complete state-of-the-art in Chapter 2), it is clear that obstacle avoidance capabilities in urban environments is a complex task and remains still as an open subject. The task is even harder when considering the dynamic nature of the environment (due to the inclusion of the temporal dimension), where in most techniques, comfort considerations are neglected. The proposed approach for obstacle avoidance in the present Ph.D. work focuses on the generation of a three step avoidance maneuver, where the curve generation aims to curvature minimization to enhance comfort.

4.1.1 Proposed approach

The developed obstacle avoidance algorithm is based on polynomial (Bezier curves) interpolation, generating a new avoidance path in real-time that will consider comfort constraints. Based in the optimal trajectory generation presented in the previous chapter, a new safe path can be created that avoids static and dynamic obstacles in conflict with the original trajectory. The new avoidance path will be generated in three phases as an overtaking maneuver (lane change, passing and returning to the original lane/trajectory). A new module is proposed to supervise the trajectory sent to the control stage, comparing the original trajectory to other road users trajectories present in the scene and, in case of conflict, generating an avoidance path in real-time. Obstacles that are either static or moving at lower constant speed than the ego-vehicle will be overtaken (introducing temporal constraints and the need for obstacle movement prediction). If at any stage the overtaking maneuver finds a new obstacle, the process starts again, avoiding the new obstacle. The following considerations were taken into account when it comes to design the proposed system:

1. The obstacle avoidance module only considers two-lane roads, with traffic traveling in one direction (the same as the ego vehicle).

2. The obstacles to be avoided are assumed to have an uniform motion profile, i.e. the obstacles are static or they are moving at constant speed (no acceleration is considered a priori).

3. The overtaking maneuver is the implemented obstacle avoidance maneuver because of structured environments and traffic laws. It has three main stages: 1) Lane change; 2) Passing; and 3) Return to the original path.

4. Comfort is considered when planning the new trajectory as in [ISO, 1997]. Lateral accelerations are computed with respect to the curvature of the path and the current velocity of the vehicle. Only the ego-path will be changed in the present approach and not the speed profile.

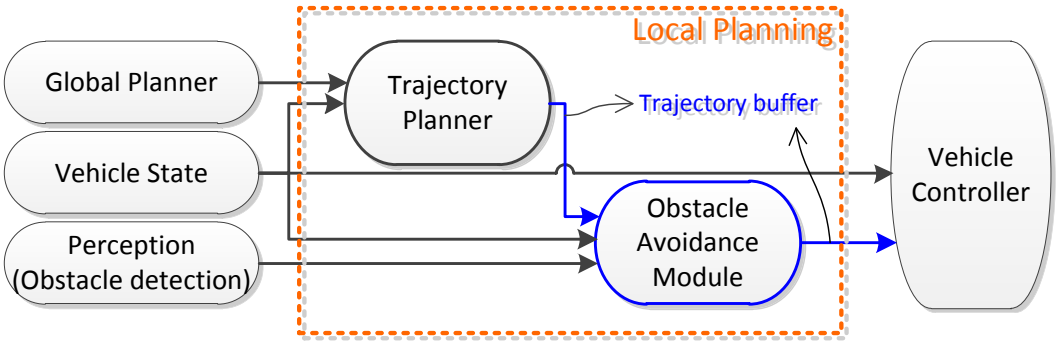


Figure 4.1: Local planning related modules: inclusion of the obstacle avoidance module

4.2 Local Planning in dynamic environments

The planning architecture presented in the previous chapter was designed for the generation of minimum curvature trajectories for optimal comfort. However, these trajectories were only able to generate paths with respect to global waypoints and therefore not flexible enough for obstacle avoidance capabilities. To do so, it is necessary to supervise the interaction of the vehicle with the environment. This is done within the framework presented in Figure 4.1, inserting a new component in the local planning architecture, capable of supervising the original trajectory sent through the communication buffer and produce, in real-time, a new avoidance trajectory if necessary.

Figure 4.1 presents the modified local planning architecture proposed for obstacle avoidance. Changes from Figure 3.4 introduce the obstacle avoidance module in between the trajectory planner and the vehicle controller. The objective is to separate the planning process in a cascade system. In a first step, the trajectory planner produces an optimal trajectory with respect to minimal curvature path and complying with comfort limits and sends it through the trajectory buffer towards the vehicle controller. In a second step, the obstacle avoidance module compares the information related to obstacle detection—given by the perception stage—and the vehicle state (proprioceptive information), performing a fast collision state check. In case no obstacles are found to be in conflict with the ego-vehicle initial trajectory, the module bypasses the information of this first trajectory. On the other hand, if a collision state is found, a new path is computed taking the vehicle to the other lane and initiating an overtaking maneuver.

The choice of decoupling the planning process in two different strategies for: 1) static environments (no obstacles, road layout known in advance); and 2) dynamic ones (static and dynamic obstacles), is because the first planned trajectory is the one used by the vehicle and will not be changed in free collision scenarios. When a future collision state has been detected, this first planned trajectory is discarded and a new trajectory is proposed to avoid the obstacle. However, the former trajectory is still generated as the optimal reference to come back as soon as the obstacle avoidance maneuver is finished. The underlaying reason is that when an obstacle avoidance is triggered, the ego-vehicle is moving to lanes with opposite traffic or, more generally, out of its optimized trajectory. This approach allows to only re-plan when needed and not all the time, improving performance.

The overtaking maneuver has been chosen for the obstacle avoidance module, achieving a general and modular approach within the functional architecture. The generality is met by defining the obstacle avoidance as an overtaking: two lane changes (one to avoid and the second to return to the original road) and a passing stage, where in case of multiple obstacles a combination

of overtaking maneuvers can be performed. Modularity is met because the trajectory planning module remains unchanged and the obstacle avoidance module acts only over the trajectory buffer (represented in blue in Figure 4.1). The main contributions of this section are:

- Real-time detection of future collision states from perception and vehicle state information.
- G^2 continuous avoidance path by implementing quintic Bézier curves.
- Real-time obstacle avoidance paths for urban environments.

4.2.1 Real-time Trajectory Modification

For detecting in advance future collision states, the ego-vehicle and obstacles future states are computed. The perception identifies the type of the obstacle is in front, its speed, dimensions, among others. It is then possible to infer the behavior of the obstacle, comparing the ego-vehicle future states against the current position of the obstacle—or future predicted states in case it has non zero velocity.

Two situations may result from the previous comparison: 1) No future collision is expected: If the result of the collision detection is negative, then the current planned trajectory remains unchanged, whether it is the original one or the new avoidance trajectory; and 2) Future collision detected: The overtaking maneuver is triggered since the trajectories of the vehicle and the obstacle are in conflict (posible collision states have been found). The collision detection is active at all times, if a new obstacle is encountered while overtaking, the obstacle avoidance module will generate a new trajectory if available. The present approach only considers lateral actions (for the ego-vehicle) and constant cruise speed (for both, the obstacle and the ego-vehicle). In case that no new trajectory is found, the control stage will be informed to adapt its speed accordingly (e.g. ACC or CACC implementations). The following subsections explain the approach in detail.

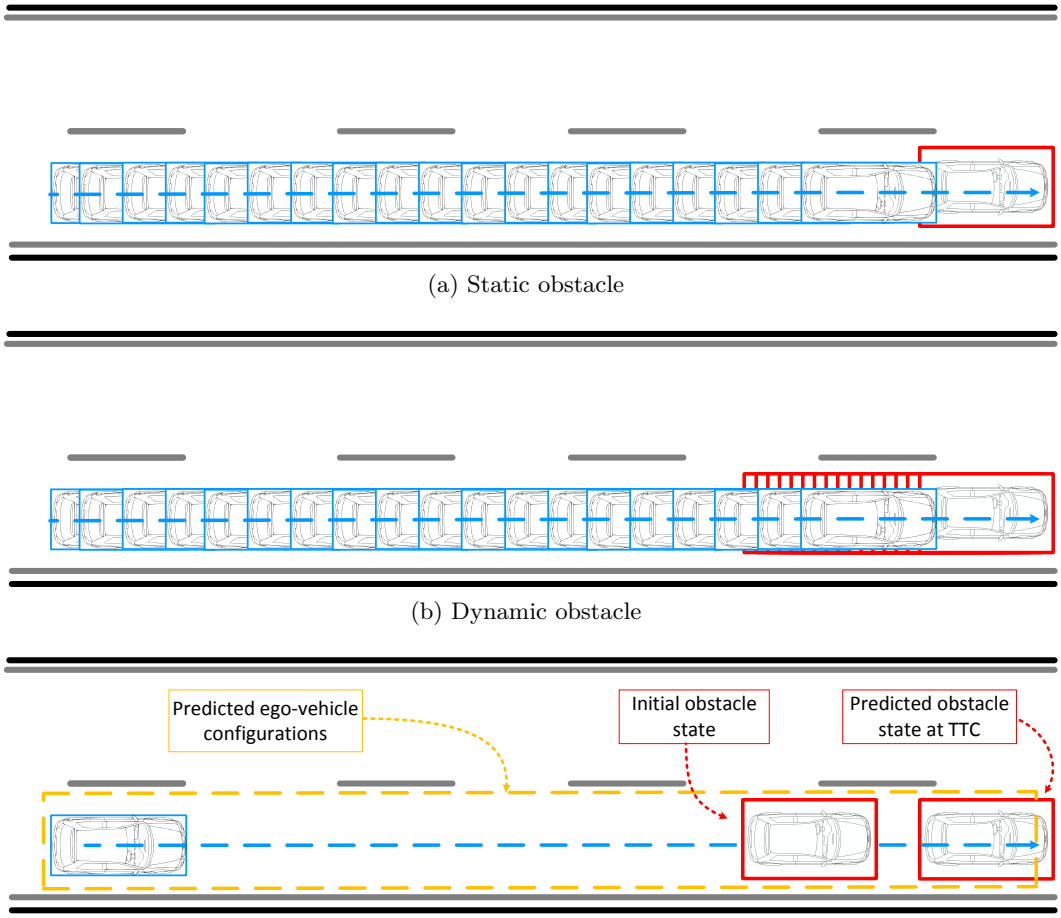
4.2.1a Detecting possible collision states

Collision states are defined as the ones where the vehicle and the obstacle cover (partially or completely) the same physical space. In robotic applications, the problem is defined as the group of manifolds in which the robot is overlapped with the obstacle [LaValle, 2006]. The common definition, explained by LaValle, is as presented in Equation 4.1,

$$\mathcal{C}_{obs} = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{O} \neq \emptyset\}, \tag{4.1}$$

where the configuration space $\mathcal{C}$ defines all the possible manifolds of the robot, $\mathcal{A}$ and $\mathcal{O}$ represent the rigid body of the robot and the obstacle region respectively, both in $\mathbb{R}^2$; and q represents the configuration of $\mathcal{A}$ (where $q \in \mathcal{C}$). The obstacle region is defined as a subset of the total configuration space, i.e. $\mathcal{C}_{obs} \subseteq \mathcal{C}$, where the obstacle and the robot have a nonempty intersection set.

With this in mind, possible collision states are defined as those $\mathcal{C}_{obs}$ regions that can be detected by predicting the obstacle and the automated vehicle (robot) behavior in $\mathcal{T}$ future states q_τ, where $d\mathcal{T} = \tau$ represents future timesteps. This means that the collision has not happened yet, but it is certain if the given trajectories of the vehicle remains identical. To compute this, the predicted configurations of the vehicle q_τ are evaluated with respect to the configurations of the obstacle. In the case where the obstacle is a static one, the multiple configurations of the vehicle $\mathcal{A}(q_\tau)$ are evaluated to the static geometry of the obstacle $\mathcal{O}$. Figure 4.2a represents the generation of both



(a) Static obstacle

(b) Dynamic obstacle

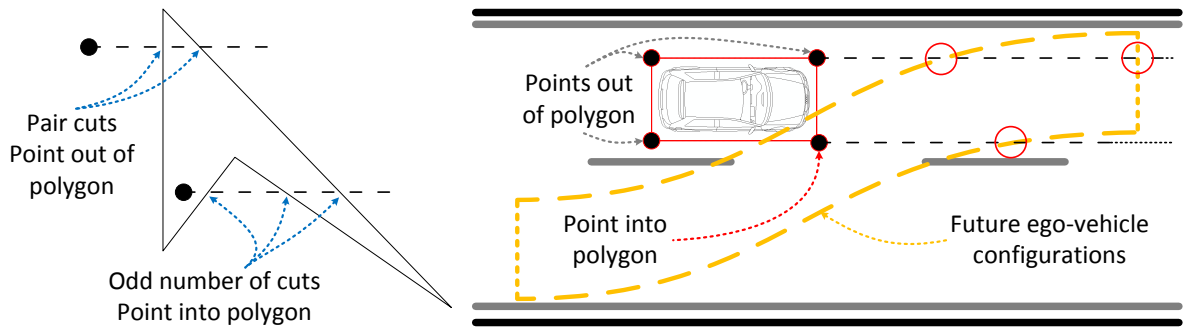
(c) Polygon intersection

Figure 4.2: Possible collision states detection. Upper two Figures show a common approach in the literature: Comparing ego-vehicle states (blue squares) against the obstacle future predicted states (red squares). The lower figure presents the proposed approach.

sets, where the vehicle's geometrical model $\mathcal{A}$ is a simple rectangular shape. The obstacle $\mathcal{O}$ is also modeled with a rectangular bounding box.

In the case where the obstacle is moving, the static assumption for $\mathcal{O}$ no longer holds. Therefore the evaluation of the trajectories is done by predicting the behavior of $\mathcal{O}$ through time. Figure 4.2b represents the case where $\mathcal{O}$ is another vehicle in the road, with non null velocity. Since another vehicle has been detected, the prediction of $\mathcal{O}(q'_\tau)$ will take constant velocities and assume a coherent driving behavior, i.e. the vehicle will remain in the lane where it is engaged (where q'_τ is the change of configurations of the obstacle $\mathcal{O}$ through time with $q' \in \mathcal{C}$). Notice that the different states of the vehicle in Figure 4.2b are more spaced than those of the obstacle, representing the speed difference between both.

However, this remains computationally expensive since all configurations of both, the obstacle and the automated vehicle need to be compared. To reduce the computation cost is possible to compare the occupancy of the obstacle and the ego-vehicle only at the presumed moment of collision, having the same result (the detection of future collision states). This translates into computing the Time To Collision (TTC) with the obstacle (see Equation 4.2), predicting its future



(a) Method example (b) Future collision check

Figure 4.3: Point into polygon example and application to collision check in automated vehicles

position only at this time horizon and then check if there is a collision configuration $\mathcal{C}_{obs}$ as in Equation 4.1. Figure 4.2c presents the proposed approach for the collision detection, where only the future configuration of the obstacle at TTC will be evaluated against the polygon formed by future configurations of the ego-vehicle. The idea is to profit from the polygonal shape of the ego-vehicle's projection over the future trajectory (the one is known a priori) and do the computation of a possible $\mathcal{C}_{obs}$ in that region which comprises all future states of the ego vehicle up to the TTC. Equation 4.2 presents the computation of the temporal prediction,

$$TTC = \frac{d_{(obs)}}{V_{(rel)}} \tag{4.2}$$

where TTC is the predicted time to collision with the obstacle, $d_{(obs)}$ is the euclidean distance from the ego vehicle to the obstacle and $V_{(rel)}$ is the relative speed between the ego-vehicle and the obstacle.

The predicted ego-vehicle configurations—over the trajectory—shape the polygon describing future ego-vehicle states. It will be then compared to the obstacle predicted state. The polygon comprises the dimensions of the vehicle throughout the known trajectory. The configuration of the obstacle at TTC is predicted to be within the lane it is engaged, meaning that a correct driving behavior is assumed where the obstacle makes no lane changes. Figure 4.2c shows the ego-vehicle initial configuration in blue, the predicted polygon describing future ego configurations in yellow and the obstacle initial and final configurations in red squares (at $t_{(0)}$ and TTC respectively). The blue square has a closer representation of the ego-vehicle than the red one used for the obstacle. This is because the perception uncertainty is included in the bounding box as a safe distance to mitigate the measurement error.

Once the above is in place, the possible collision states can be computed. The problem can be posed as a point-into-polygon problem [Shimrat, 1962], [Hacker, 1962]. Figure 4.3a presents the general approach over two points, one on the outside and one on the inside of a simple polygon.[1] The approach shows that is possible to determine if a point is inside or outside a polygon by projecting a line from the point to the infinite (a value bigger than the coordinates of the polygon will suffice) and counting the times the line intersects the polygon. If the number of intersections is pair, the point will be labeled as an outsider (see upper black point in Figure 4.3a). On the other hand, if an odd number of intersections is found, the point is within the polygon (lower point within the polygon in Figure 4.3a).

[1]These are 2D polygons formed of straight, non-intersecting segments that form a closed shape.

With this in mind, it is possible to check future collision states since the points describing the bounding box of the obstacle (red in Figure 4.2c) are compared against the yellow polygon to see if the obstacle is inside (verify if the condition from Equation 4.1 is true, i.e. $\mathcal{C}_{obs}$ is non empty). Figure 4.3b shows an example of how this is implemented in the proposed approach. Following the point-into-polygon method, only one of the points has an odd number of intersections—intersections are highlighted in red—meaning that it lies within the polygon (the lower-right point of the bounding box) and also meaning that the obstacle is in conflict with the trajectory of the ego-vehicle and its path needs to be reconsidered. Moreover, to avoid unnecessary computations of collision regions and speed up the detection process, obstacles with a negative TTC (meaning that the obstacle is getting away from the vehicle) are not taken into account. The reason is because a collision is highly unlikely, therefore reducing the search only to those obstacles that might generate a collision and thus optimizing the real-time response. This implementation of the point-into-polygon algorithm permits a fast future collision detection and improves significantly the real-time response of the obstacle avoidance module.

Algorithm 4 Future collision states detection

1: **Start**
2: **Read** algorithm properties
3: **Get** local trajectory information
4: **Get** vehicle state
5: **Get** obstacle detection information
6: **Set** the polygon associated to the ego-vehicle future configurations
7: **Set** $point_in_polygon = false$
8: **Compute** the TTC
9: **if** $TTC > 0$ **then** Check collisions
10: **Compute** obstacle configuration at $t = TTC$
11: **for** $k = 0; k < size(obstacle_bounding_box); k++$ **do**
12: **for** $i = 0; i < size(polygon); i++$ **do**
13: **Count** $number_of_cuts$
14: **end for**
15: **if** $number_of_cuts\%2 == 0$ **then** $point_in_polygon = false$
16: **else** $point_in_polygon = true$
17: **Break** for ▷ The ego-vehicle has a future collision state
18: **end if**
19: **end for**
20: **end if**
21: **Return** $point_in_polygon$
22: **End**

Algorithm 4 shows the proposed implementation for the detection of future collision states. Lines one to seven initialize the algorithm. In line eight the TTC is computed. Line nine prevents to compute future collision states with obstacles getting away from the ego-vehicle (this results in a relative velocity $V_{rel} < 0$ and a $TTC < 0$) which are highly unlikely to be in conflict with the ego-vehicle in future configurations. Lines ten to 19 implement the point-into-polygon algorithm, first computing the predicted configuration of the obstacle at TTC (see, line ten). Each point of the obstacle's bounding box is verified against the whole polygon (lines 11 to 14). If at any moment a point is found to be within the polygon, the algorithm stops automatically and returns

true for an expected collision state with the obstacle in hand (lines 15 to 17). This procedure will be done for all obstacles detected by the vehicle.

Clearly, if an obstacle is detected to be in conflict with the current trajectory, a decision should be issued of how to avoid this obstacle. In the following, the interest subject is to evaluate how to generate a path in real-time when the chosen decision is to swerve and , thus, to avoid the obstacle (the decision could also be to avoid the obstacle by modifying the speed profile—instead of creating a new path—and follow the obstacle as in CACC implementations [Milanés et al., 2014b]).

4.2.1b Parametric curves for obstacle avoidance

Once the ego-vehicle has spotted an obstacle and determined that this obstacle is in conflict with its trajectory, the decision of an avoidance maneuver may be taken. If so, the vehicle initiates an obstacle avoidance process defined in three main steps: an avoidance lane change, passing the vehicle (normally on the left for right handed driving countries) and a lane change to return to the original lane. This *"three steps"* proposed approach is normally known as an overtaking maneuver and profits from the structured shape of the scenario. Each step needs to be tangent and curvature continuous with respect to past and future segments. To reach this level of continuity, the implementation of G^2 continuous paths is the most attractive solution, given that this level of geometric continuity ensures curvature continuity and avoids the lack of flexibility of C^2 parametric continuous curves.

The chosen curves for this task are Bézier curves. These curves have important properties which, applied to real-time trajectory generation and obstacle avoidance, can greatly improve the computation time in complex scenarios. In Sections 3.4.1 and 3.4.3b the Bézier curves where chosen due to its flexible and fast curve generation characteristics, where also some of the properties of the Bézier curves where highlighted. In contrast to 3rd and 4th degree curves implemented for smoothing the singular global points and paths within roundabouts in the previous chapter, 5th degree curves will be applied for the generation of lane change paths. This configuration of the Bézier curve permits G^2 continuity at joints with other curved and straight segments, while also allowing an inflection point in the curve and fast convergence to the minimal curvature path, reducing lateral accelerations and increasing comfort.

The necessary mathematical expressions for quintic Béziers generation is presented in Section 3.4.3b, where the most important property is the capacity of a quintic Bézier to adapt its curvature at the beginning and at the end of the curve. This is presented by Equation 3.23 where the value of γ (the angle between the first and second control points vectors) and the control points inter-distance d determine the initial (for the parameter $t = 0$) and ending curvature (for the parameter $t = 1$) for the Bézier curve. Figure 4.4 shows the generation of 5th degree Béziers applied to obstacle avoidance, where the length L_{Lc} represents the longitudinal distance given to perform the lane change and W_{Lc} the lateral displacement. The control points are presented in blue and the distances between them are represented by d_1, d_2, d_3 and d_4. The convex-hull property forces the curve to stay within road constraints since the beginning and ending vector of the curve are tangent to the road direction (and so are the control points that describe the curve).

To optimize the curve in real-time, the process needs to be as less time consuming as possible while providing the best possible control points configuration. With this in mind, different numerical optimization configurations were tested to find the one with the best results.[2] Figure 4.5 shows the generated curves after an optimization process for each case described in Table 4.1. The test aims to provide some insight on the behavior of the Bézier curves and to clarify the choice made

[2]Numerical optimization is preferred since no algebraic solution exists for n^{th} degree polynomials, where $n \geq 5$.

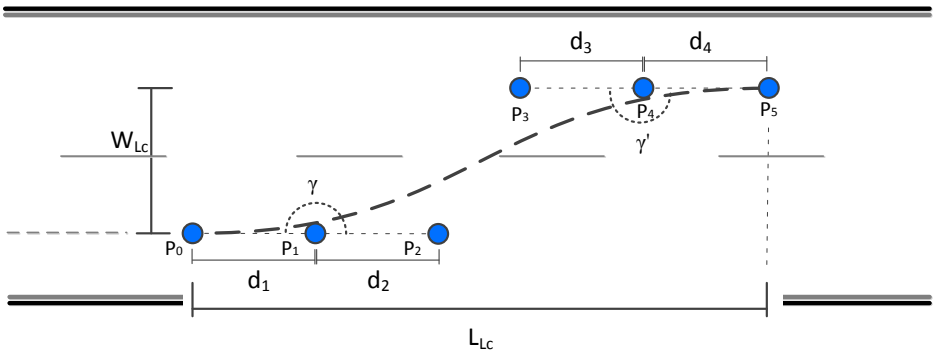


Figure 4.4: Fifth degree Bézier curve and its control points applied to obstacle avoidance

with respect to the optimization parameters (OP in Table 4.1) and the iteration configurations (IC in Table 4.1). Two optimization parameters were chosen for initial testing: maximum curvature (where the optimization selects the curve with the lowest maximum for its curvature throughout the whole curve) and maximum curvature derivative (where the optimization selects the curve with the lowest maximum for its curvature derivative throughout the whole curve).

The optimization process is done by iterating over the control points inter-distances. Two iteration configurations were tested. First, a complete use of all inter-distances was implemented, where the algorithm iterated through each of the control points inter-distances, where d_1, d_2, d_3 and d_4 (distances described in Figure 4.4) went from 0 to L_{Lc} in steps of $L_{Lc}/300$ (which gives steps of $0.1m$ for the example in Figure 4.5). In addition, control points were forced to stay within the range of the search area, described as inequations 4.3 and 4.4. Results are shown in Figure 4.5, where curves are presented in red and black.

$$0 < d_1 + d_2 < L_{Lc} \tag{4.3}$$

$$0 < d_3 + d_4 < L_{Lc} \tag{4.4}$$

The second iteration configuration was to equalize the first two and last two distances (i.e. $d_1 = d_2$ and $d_3 = d_4$) and only iterate d_1 and d_3. As in the previous case, the iteration goes from 0 to L_{Lc} in steps of $L_{Lc}/300$, where control points are also forced to stay within search limits, i.e. complying with inequations 4.3 and 4.4. Results are shown in Figure 4.5, where curves are presented in green and magenta. This last configuration provides a lower time of search for the best profile in both cases (minimization of the curvature or the curvature derivative), however is evident that provides a less complete solution than the exhaustive search which explores all inter-distances combinations.

Figure 4.5a presents the plotted curve for each of the optimization configurations, showing candidate approaches for lane change maneuvers in obstacle avoidance applications. Figure 4.5b shows the curvature profile for each approach, where the lowest curvature is described by the red curve. The derivative of the curvature (mostly related to lateral jerk) is presented in Figure 4.5c. It shows a smooth behavior and low values for most of the curves, where the black and the magenta curves have the lowest curvature derivative.

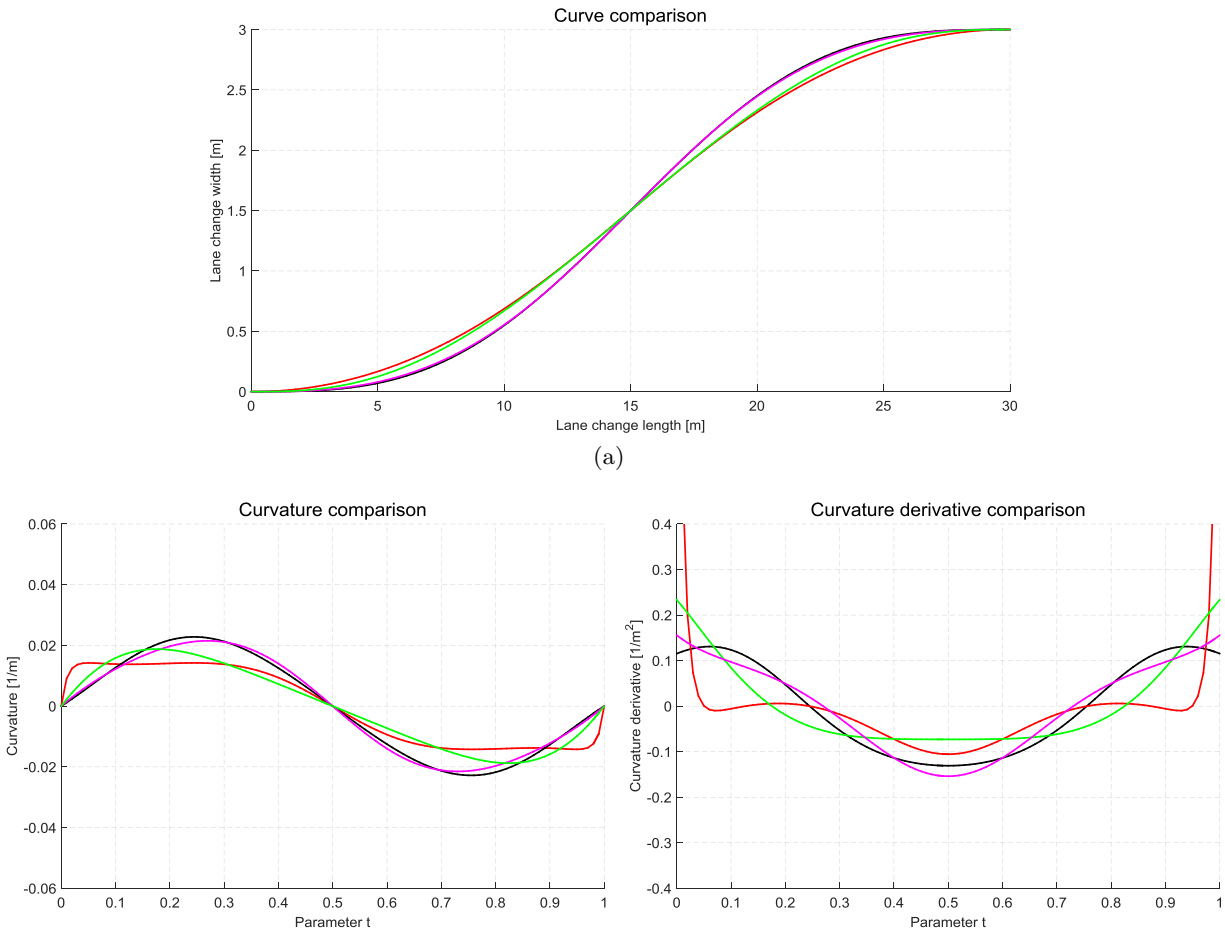


Figure 4.5: Different optimization configuration for quintic Béziers

Table 4.1: Optimization and iteration parameters for quintic Béziers

OP \ **IC**	**Complete search** $d1$ through $d4$	**Simplified search** $d1 = d2$ and $d3 = d4$
Curvature	Red curve: Has the lowest curvature value throughout the curve but presents fast curvature changes at the beginning and the end, i.e. values of $1.48/m^2$ in the derivative of the curvature.	Green curve: Gradually distributes the curvature throughout the length of the curve. The maximum curvature and maximum derivative are low (less than $0.02/m$ and $0.2/m^2$ respectivelly).
Curvature derivative	Black curve: The curvature is equally distributed over the curve (best curvature derivative profile), however its maximum curvature is higher than other configurations.	Magenta curve: Has a slower response than the green curve, the maximum curvature is higher than the green and red curves.

From the comparative in Table 4.1 and Figure 4.5, the configuration producing the green curve is the one which fits better with the present approach for obstacle avoidance with comfort considerations. It minimizes the curvature of the Bézier as much as possible by iterating through two variables representing the control points inter-distance. It is chosen over the red curve configuration because this one shows high curvature derivative values which could create discomfort with a fast change over the curvature. Also by performing a simplified search (reducing in two the amount of variables) the algorithm finds a solution faster than with the complete search. Black and magenta approaches served to rule out possible optimal choices from the derivative of the curvature point of view. The results are promising, but the lowest curvature (the magenta curve) is not lower than the green approach, where once more the green approach is best fit since the link to comfort is more direct with curvature than with curvature derivative.

The behavior of quintic Bézier curves in lane change scenarios (in Figure 4.4) give an accurate insight on how to optimize the trajectory generation process in real-time. The same scenario was repeated from $L_{Lc} = 5$ to $L_{Lc} = 150$ with similar results. Once the method to compute lane changes paths in real-time is established, the decision to generate a new avoidance path is executed as explained in the following section.

4.2.2 Obstacle avoidance/overtaking

The obstacle avoidance problem can be broken down in smaller problems and assess each situation individually. For automated driving, this can be performed as an overtaking maneuver, as in [Naranjo et al., 2008], where a first lane change avoids a future collision with the obstacle. Then, a passing section considering the length and speed of the obstacle allows the ego-vehicle to overtake and a second lane change returns the vehicle to the original lane. The maneuver involves quick decision-making to plan trajectories in real time, which translates into steering wheel actions, considering the current velocity of the vehicle, producing a comfort trajectory that avoid the obstacle. The maneuver is divided in three phases as depicted in Figure 4.6, where the conflicted trajectory is shown in red and the new one is presented in blue. These phases are described as follows:

4.2.2a Avoidance

This first phase plans the avoidance path once an obstacle is found to be in conflict with the predefined trajectory. The starting point of the curve (P_0) is located over the previous trajectory in front of the ego-vehicle (considering the look-ahead distance in the control stage), with tangent and curvature constraints that mimic the ones in the conflicted path to keep a G^2 continuous profile. The last control point (P_n) of the Bézier curve will be defined by the configuration needed to avoid the obstacle. Figure 4.6a presents this first phase going from a local point A to point B, where the point P_n will be placed to the left of the overtaken obstacle, considering the ego-vehicle's dimensions, the obstacle's dimensions, the relative distance and speed between them. This new trajectory is also compared to detect possible conflicts with other obstacles (as in Section 4.2.1a) and take safety measures if so. The obstacle to overtake is assumed to have a coherent movement with respect to the lane in which it is engaged, however lateral and longitudinal safe distances (SD_{lon} and SD_{lat} respectively) are set to increase safety and cope with sensors and perception uncertainty. Once the initial and final control points have been set, the curve is optimized and generated as in Section 4.2.1b.

To define the position of P_n for the avoidance phase, the rear part of the obstacle (the closest to the ego-vehicle) is located in the original trajectory (in red in Figure 4.6a). The point where the

obstacle is located in the trajectory is defined as $T_{(obs)}$ (with position x,y and orientation θ, see the red dot in Figure 4.6a). In the case where the obstacle is moving, the future predicted configuration of the obstacle is taken at TTC (i.e. the back part of the vehicle in the future configuration is taken). Equation 4.5 gives the mathematical relationship between where the obstacle is found in the conflicted trajectory and the placing of P_n,

$$\begin{aligned} P_{n_x} &= T_{(obs)_x} + \left(\frac{W_{ego}}{2} + LO_{obs} + SD_{Lat}\right)\cos\left(T_{(obs)_\theta} + \frac{\pi}{2}\right) - (L_{ego} + SD_{Lon})\cos\left(T_{(obs)_\theta}\right) \\ P_{n_y} &= T_{(obs)_y} + \left(\frac{W_{ego}}{2} + LO_{obs} + SD_{Lat}\right)\sin\left(T_{(obs)_\theta} + \frac{\pi}{2}\right) - (L_{ego} + SD_{Lon})\sin\left(T_{(obs)_\theta}\right) \end{aligned} \tag{4.5}$$

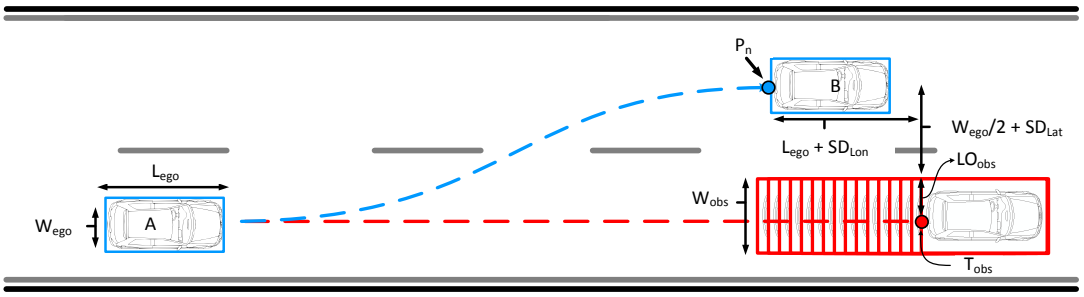
where P_{n_x} and P_{n_y} are the Cartesian coordinates for the final configuration at point B of Figure 4.6a and n represents the order of the curve (i.e. $n = 5$ for the present approach). The dimensions of the ego-vehicle are represented by L_{ego} and W_{ego}; the location of the obstacle is determined by the first term of the equation, i.e. $T_{(obs)}$, within the original trajectory. The lateral occupancy of the obstacle in the lane is define as $LO_{(obs)}$ (since the obstacle might have a lateral displacement, see Figure 4.6a). Safe distances are defined as SD_{lon} and SD_{lat}, set at $0.5m$ and $0.3m$ respectively for the present approach. The second and third term describe de lateral and longitudinal displacement for P_n respectively. Finally, $T_{(obs)_\theta}$ represents the orientation of the closest point in the original trajectory with respect to the obstacle.

4.2.2b Passing

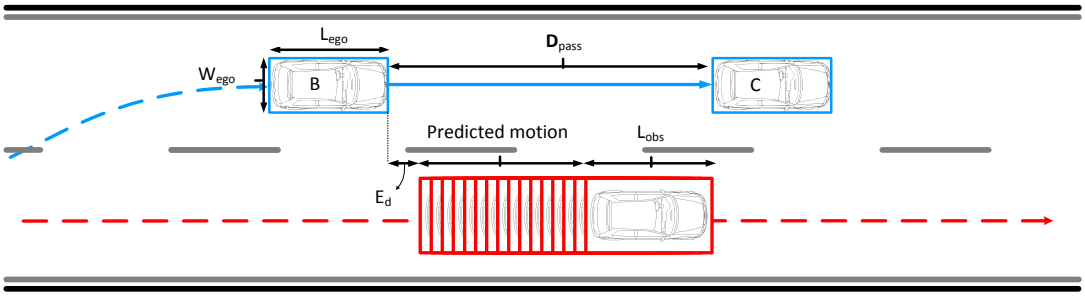
The second phase starts once the ego-vehicle can measure the length of the obstacle. Since the visible area of the obstacle is only the rear part (at the beginning of the avoidance maneuver), the passing path will be computed when the information of the length of the front obstacle is known. The planner assumes that this information is available when the vehicle has already traversed half of the avoidance path (half of the trajectory generated in the first phase). In case when this information is not yet available, the planner assumes a length of 3m (as commercial vehicles) and starts the middle phase with this assumption. If the obstacle length changes through time (a better detection is available), the planner will update the passing trajectory. Figure 4.6b presents the passing strategy when the obstacle is moving. The proposed approach considers constant velocity for the obstacle and computes its predicted motion. With this information and the obstacle's dimensions is possible to set the length of the passing trajectory, which is set for the ego-vehicle to completely pass the obstacle, i.e. from position B where the vehicle is still behind the obstacle but on the left lane, to position C where the ego-vehicle has cleared the obstacle and is now in front of it, still on the left lane. The previous trajectory is kept in a buffer and serves as reference for the passing stage. The distance between position B and position C of Figure 4.6b is presented in Equation 4.6,

$$D_{pass} = 2L_{ego} + L_{obs} + E_d + \frac{L_{obs}V_{obs}}{V_{rel}} \tag{4.6}$$

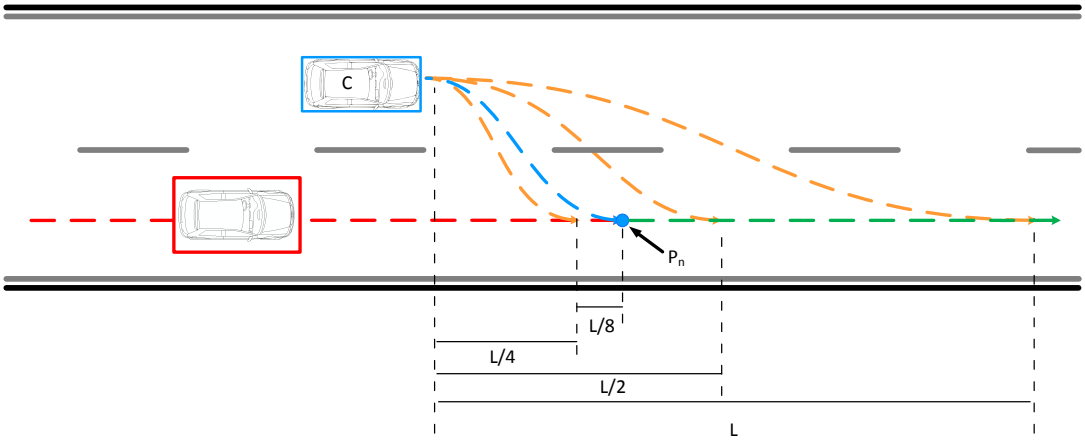
where D_{pass} is the needed distance to clear the obstacle. The vehicle and obstacle lengths are represented as L_{ego} and L_{obs} respectively. The terms $2L_{ego} + L_{obs} + E_d$ appear since the ego-vehicle is to completely clear the obstacle, so it suffices to account for twice the ego-vehicle length and once the obstacle length, where the E_d term handles possible distance errors due to movement or safe distances considerations of the previous avoidance stage. The third term compensates the distance the ego-vehicle needs to pass an obstacle in movement. It will be zero when the obstacle is static $V_{obs} = 0$.



(a) Avoidance



(b) Passing



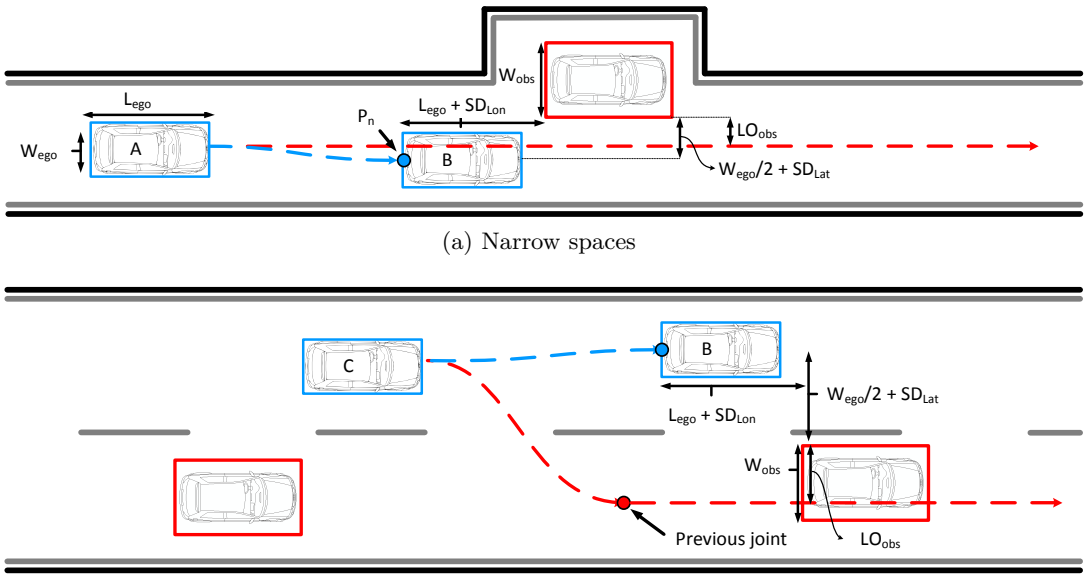
(c) Return to the original path

Figure 4.6: Overtaking maneuvers in three stages

4.2.2c Return to the original path

This phase allows the ego-vehicle to return to the right lane. Here, the constraints when setting the control points will ensure a G^2 continuous path. The first control point P_0 is set at the end of the passing trajectory (the ego-vehicle configuration at C, the end of the passing stage). However, the last point of the curve is unknown. To find it, it is possible to search within candidate trajectories that will bring the ego-vehicle to the original lane as fast as possible, but complying with lateral acceleration limits (to keep comfort).

Since the vehicle has to return to the right lane (described by the original trajectory), the last control point of the curve (P_n) will be searched within this initial path. The idea is to find the closest point the vehicle can re-join the original trajectory without exceeding comfort limits



(a) Narrow spaces

(b) Successive obstacles

Figure 4.7: Special considerations in obstacle avoidance

in [ISO, 1997]. The search is done within a horizon distance set at $100m$, where the optimization of the trajectory permits to have the optimal curve for each one of the test distances, obtaining also the lower lateral acceleration for the assumed constant velocity of the ego-vehicle.

Figure 4.6c shows the original path (in red), the generation of each of the test curves (in orange) and the chosen path to return to the original lane (in blue). The module implements a bisection approach to find the optimal configuration of (P_n) which gives the best curve (closest to comfort limits and which takes as fast as possible the vehicle back to the original lane). Orange curves (Figure 4.6c) appearing on the right side of the blue curve (i.e. at distances L and $L/2$) are those curves that are within comfort constraints but take longer to go to the original path. The orange curve on the left side of the blue curve (i.e. P_n at distance $L/4$) are those curve that take the vehicle to the original path faster than the blue curve, but surpass comfort limits. The bisection approach finds a proper trajectory within six to seven iterations and an error of $2m$ for the location of P_n over the original path. Once the path is created, the planner joins the blue and green stretches (end of the avoidance and the original path respectively, see Figure 4.6c), for the ego-vehicle to continue in the optimal trajectory if is not in conflict with other obstacles. Moreover, in the cases where the vehicle travels in low speeds (i.e. lower than $3m/s$), the kinematics of the vehicle are also taken into account and is the maximum curvature of the vehicle the constraint considered as to plan the curve when returning to the original path (see Equation 3.8).

4.2.2d Considerations on overtaking maneuvers

The proposed approach adapts the trajectory to obstacles, without intervening in the generation of the optimal trajectory. It is also adaptable since it addresses most situations within the constraints posed in Section 4.1.1, enabling obstacle avoidance via overtaking maneuvers. However, there are still open considerations in the overtaking, i.e. on which side to overtake (left or right) and how to manage misaligned and/or successive obstacles.

Figure 4.7 presents graphic examples of these considerations to apply the approach to different

situations, normally present in urban environments. Figure 4.7a presents one common situation where vehicles are allowed to park in the side of the road, but slightly blocking the road. This configuration is common in European countries, where roads are small and there is not much room for parking spaces. The obstacle avoidance module has then to decide whether to overtake on the right or on the left. The decision is taken by assessing the space where the ego-vehicle is able to navigate, at each side of the obstacle. In Figure 4.7a, is clear that the ego-vehicle can and will overtake on the right (assuming the obstacle is static). In contrast with the example in Figure 4.6a, the obstacle is not engaged in the lane that the ego-vehicle travels, but the original trajectory remains conflicted since a future collision has been detected (original trajectory in red and avoidance trajectory in blue, see Figure 4.7a).

Although the consideration of multiple obstacles is out of the scope of this Ph.D. work, the obstacle avoidance module is capable to address scenarios where subsequent obstacles are in the same lane while overtaking. Figure 4.7b presents an example with two static consecutive obstacles in the same lane. The first obstacle has already been passed, the ego-vehicle plans its path back to the original paths as in Section 4.2.2c. However, this path is in conflict with the second obstacle. The ego-vehicle then restarts the avoidance process, where position C in Figure 4.7b transforms into the starting position of the avoidance phase (creating a new overtaking maneuver). The P_n point is planned as in Section 4.2.2a, discarding the conflicted trajectory in red and creating a new safe trajectory, depicted in blue (see Figure 4.7b). The obstacle avoidance module also considers the lateral displacement of this second obstacle and generates the new path accordingly (since the second obstacle is more to the left than the first one), given that the ego-vehicle still clears the curb.

Algorithm 5 presents the general structure within the proposed obstacle avoidance module. Lines one to five initialize the algorithm. Lines six implements Algorithm 4 to detect future collision configurations with obstacles. If no obstacle is in conflict with the initial trajectory, the obstacle avoidance module only bypasses this initial trajectory and stays in supervision mode (lines 38 to 40). If an obstalce is in conflict with the trajectory, the obstacle avoidance is initiated. The three phases of the overtaking maneuver are well described in lines nine, 19 and 28.

The avoidance maneuver will compute the side to overtake (line ten). The optimization of the curve with respect to P_0 and P_n and the generation of the curve is performed in lines 12 and 13. If the avoidance phase has not enough space to avoid the obstacle or if another vehicle is in conflict with the avoidance maneuver (lines 14 to 16), the control and decision stages (see Figure 3.1) are informed in order to take safety actions and generate a new strategy (e.g. longitudinal actions can be also taken into account at this point). If the new path is clear from conflicts with obstacles, it is sent to the control stage through the communication buffer (line 17).

When the ego-vehicle has traveled through half of the trajectory produced in the avoidance phase, the passing phase is initiated. Here the passing distance is the most important value to compute (lines 19 to 21). As in the previous phase, if in the passing phase obstacles are found to be in conflict with the new path, safety measures are taken and a new strategy is requested. In case the path is free, the generated trajectory is sent through the communication buffer to the control stage.

The last phase is initiated once the ego-vehicle completed the avoidance path. This is, the avoidance and passing phase have already been planned, the ego-vehicle is located at the end of the avoidance phase (Figure 4.6b) and now the third phase is planned in advance. The returning phase is presented in line 28, P_0 is set and P_n is found by a bisection approach in line 29. At each iteration of the bisection approach the optimization of the Bézier curve is performed, as to find the minimal curvature for each case; once P_n is selected, the curve is generated in line 30.

Line 31 checks possible collision with other obstacles in the returning path. If a future collision is detected, the avoidance phase is initiated and the process stars once again (lines 32 and 33). In case no collision is foreseen, the overtaking maneuver has finished (lines 34 and 36) and the module returns to a supervision mode.

4.3 Conclusion

This chapter presented a novel real-time path planning algorithm for obstacle avoidance in urban environments. It complements the optimal trajectory created in the previous chapter by adding obstacle avoidance capabilities without modifying the normal operation of other modules. The main contribution is the conception of a local planning architecture that allows the vehicle to follow the optimal trajectory and avoid obstacles, modifying the trajectory only if necessary. The obstacle avoidance module is in cascade with the trajectory planner as proposed in Figure 4.1, where the avoidance module intervenes over the communication buffer, modifying the path if needed and sending it to the control stage in real-time.

To determine whether to modify the trajectory or not, the detection of possible collision states is necessary. The prediction of future obstacle configurations was computed up to a time horizon, defined by the TTC with the ego-vehicle, which allowed to have a comprehensive scenario of the position of each obstacle and to determine if a collision is expected. The proposed approach implemented a point-into-polygon algorithm to determine if the points describing the obstacles at TTC are inside the polygon formed by the future trajectory of the ego-vehicle, and if so, a future collision state is then detected.

Quintic Béziers were implemented to produce a G^2 continuous path, based on their flexibility, advantageous properties and fast computation characteristics. The creation of the new avoidance trajectory—in real-time—relied over the properties of Bézier curves, to reduce the computation time, simplifying the search for the optimal curve. The optimization approach seeks to minimize the curvature through the path, while at the same time reduce as much as possible the computation time. The proposed configuration of the quintic Bézier assures the comfort while traveling in curved segments, since the minimization of the curvature is directly linked to the minimization of lateral accelerations (see Equation 3.18).

The proposed solution for obstacle avoidance in structured and dynamic scenarios is based on overtaking maneuvers, decoupled in three main stages: avoidance, passing and returning to the original path (as in [Naranjo et al., 2008]). These allow the automated vehicle to pass obstacles in conflict with its initial path and then return to the right lane as dictated by French driving law (where vehicles should drive on the right lane and only use the left one to overtake slower vehicles). The division of the maneuver in three stages allows the ego-vehicle to react, in real-time at every stage of the overtaking.

Special considerations with respect to obstacle avoidance have also been proposed to generalize, as much as possible, the approach in urban environments. Situations like small roads with vehicles parked on the side of the road and successive obstacles are common in these kind of environments. The proposed approach can address these situations with the same overtaking in three stages approach, by applying heuristics in the placing of the last control points for the avoidance phase.

Algorithm 5 Obstacle avoidance

```
1: Start
2: Read algorithm properties
3: Get local trajectory information, vehicle state, obstacle information
4: Set the polygon associated to the ego-vehicle future configurations
5: Set point_in_polygon = false, avoiding_obstacle = false
6: Implement Future collision states detection                          ▷ Algorithm 4
7: if point_in_polygon == true or avoiding_obstacle == true then Avoidance= true
8:     Set avoiding_obstacle = true
9:     if Avoidance then
10:        Compute lateral distances and the avoidance side (left or right)
11:        Set P_0 and P_n
12:        Generate the path                                            ▷ Curve optimization
13:        Implement Future collision states detection                  ▷ Algorithm 4
14:        if Not enough space or point_in_polygon == true then
15:            Inform control stage for safety actions
16:            Inform decision stage requesting new strategy
17:        else send the trajectory in the buffer to the control stage
18:        end if
19:    else if Passing then
20:        Set D_pass
21:        Generate the path                                            ▷ Curve optimization
22:        Implement Future collision states detection                  ▷ Algorithm 4
23:        if Not enough space or point_in_polygon == true then
24:            Inform control stage for safety actions
25:            Inform decision stage requesting new strategy
26:        else send the trajectory in the buffer to the control stage
27:        end if
28:    else Return to original path
29:        Set P_0 and Find P_n                  ▷ Bisection and curve optimization
30:        Generate the path
31:        Implement Future collision states detection                  ▷ Algorithm 4
32:        if point_in_polygon == true then
33:            Go to: line 13 and restart the Avoidance phase
34:        else send the trajectory in the buffer to the control stage
35:            Set avoiding_obstacle = false
36:        end if
37:    end if
38: else if point_in_polygon == false and avoiding_obstacle == false then
39:    Bypass the original trajectory                                   ▷ No avoidance maneuver
40: end if
41: End
```

Chapitre

Validation et applications

Below is a French summary of the following chapter "Validation tests and applications".

Ce chapitre présente les tests de validation en simulation et sur des plate-formes réelles de la méthode de planification de trajectoire proposée dans cette thèse. Une application de contrôle partagé et d'arbitrage basée sur notre architecture fonctionnelle est présentée et validée en simulation. Le confort et la génération des trajectoires en temps réel sont les critères de validation les plus importants. L'approche proposée a été simulée sur Pro-Sivic et testée dans des environnements urbains réels. Différentes configurations routières, virages et ronds-points à deux voies ont été simulés. Le système interagit avec d'autres acteurs de la route, en évitant les collisions futures lorsqu'elles sont détectées. Les tests sur des véhicules réels ont été effectués dans les circuits de l'Inria à Rocquencourt, en France. Différents virages et ronds-points sont inclus dans l'itinéraire. La plate-forme réelle est un véhicule urbain électrique appelé Cyberbus. Des manœuvres d'évitement des obstacles ont également été effectuées, à des faibles vitesses, en validant l'approche.

Les applications de contrôle partagé et d'arbitrage ont également été testées dans des scénarios simulés. L'idée est de garder le conducteur dans la boucle de contrôle, d'évaluer sa performance et d'intervenir en cas de danger. L'ADAS proposé met en œuvre les stratégies de planification décrites dans les chapitres trois et quatre comme guide pour évaluer le comportement optimal du véhicule dans différentes situations. La génération de limites de conduite, identifiées par les limites cinématiques et dynamiques du véhicule, crée une zone de conduite autorisée au conducteur. L'application proposée intervient uniquement lorsque cela est nécessaire, ce qui permet le partage du contrôle avec le conducteur et le maintien de la sécurité des passagers.

Chapter 5

Validation tests and applications

This chapter presents the validation tests and applications for the proposed functional architecture in this Ph.D. thesis. Modularity, comfort and real-time behavior are the most important design criterion. The proposed approach has been simulated and tested in real urban environments. Different road configurations, turns and two-lane roundabouts with different radii were simulated. Static and dynamic obstacles are present and so the system interacts with other road actors, avoiding future collisions when detected. Second, the real scenario is located at INRIA facilities in Rocquencourt, France. Different turns and roundabouts are included in the itinerary. The real platform is an electric urban vehicle called the Cyberbus. Obstacle avoidance maneuvers were also conducted, at low speeds, successfully validating the approach.

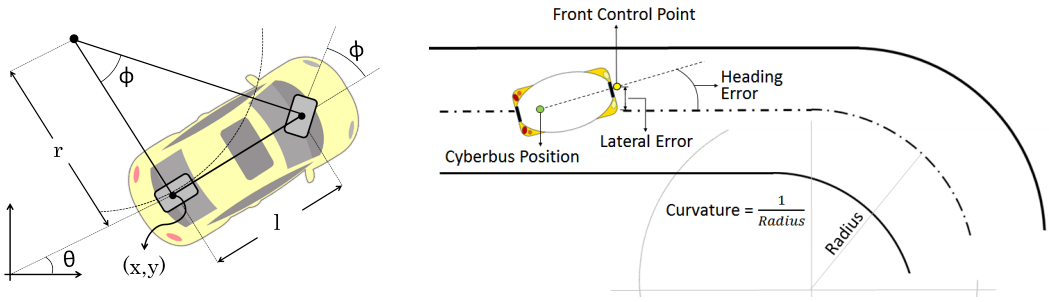
Shared control and arbitration applications have been tested in simulated scenarios. The idea is to keep the driver in the control loop and addition the system supervision over drivers' knowledge and skills with respect to the driving task. The proposed ADAS implements the planning strategies described in Chapters 3 and 4 as a guideline to assess the optimal behavior of the vehicle in different situations. The generation of driving boundaries, identified by the vehicle's kinematic and dynamic limits, create a driving area where the human can safely operate the vehicle.

The proposed application is able to assist the driver only when needed, establishing a communication bridge between both decision-makers and smoothly sharing the control in the driving task. This chapter is structured as follows: Section 5.1 describes the tests conducted to validate the functional architecture in simulated urban environments, as well as the simulation tool and the development software implemented. Once validated in simulations, the system was tested in real environments at INRIA facilities. The architecture was embedded in the Cyberbus to demonstrate the validity of the approach in real urban scenarios, detailed in Section 5.2. Shared control applications are then explained in Section 5.3, showing the modularity of the planning architecture and establishing the link between the trajectory generation and driver-in-the-loop applications. Finally, Section 5.4 provides a conclusion for the chapter.

5.1 Validation tests—simulations

This section presents the validation tests, in simulated environments. The software environment used for the implementation is RTMaps[1], a real-time tool, designed for fast and robust implementation in multi-tasks and multi-sensor data. It is based on a multi-thread structure, allowing easy adaptability to either simulated or real platforms.

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



(a) Bicycle model. (b) Error computation for the control stage.

Figure 5.1: Vehicle model and error computation.

Two simulation environments where implemented for initial test of the functional planning architecture: Pro-SiVIC[2] and Matlab. Pro-SiVIC is a software environment for the assessment of the robustness and reliability of on-board sensors. It is capable of reproducing multi-sensorial environment, considering real vehicle parameters such as inertia, steering wheel response, lateral acceleration with yaw angles, among others. It is compatible with the used software environment, i.e. RTMaps. Experiments were validated using Pro-SiVIC implementing a virtual vehicle in urban scenarios, including roundabouts.

5.1.1 Vehicle model

For simulation purposes, a vehicle model was developed for initial testing on Matlab/Simulink. Considering INRIA experimental platform, where maximum speed is limited up to 5m/s, the so-called dynamic bicycle model was used [Rajamani, 2011]. It provides good results for low speeds vehicle simulation. Vehicle control is out of the scope of this work but some notions about the current algorithms implemented on the INRIA architecture are provided in this section.

The differential equations, describing the movement in a Cartesian plane (see Figure 5.1a), are as follow:

$$\begin{aligned} \dot{x}(t) &= \frac{dx(t)}{dt} = V_t \cos(\theta(t)) \\ \dot{y}(t) &= \frac{dy(t)}{dt} = V_t \sin(\theta(t)) \\ \dot{\theta}(t) &= \frac{d\theta(t)}{dt} = \frac{V_t}{L} \tan(\phi(t)) \end{aligned} \tag{5.1}$$

where θ is the orientation angle with respect to plane x, y, ϕ is the steering angle of the front wheel, L is the wheel base and V_t is the longitudinal speed. The longitudinal behavior is modeled as a second order function already tested in real vehicles (Cybercars) in [Milanés et al., 2014a]. The longitudinal controller implemented is a Proportional-Derivative controller (PD), as in [González and Perez, 2013].

Considering low speed at turns, the wheel slipping and the forces transferred between wheels of the same axle track are approximated to zero. Thus, we assume that the maximum feasible curvature is when the turning angle of the wheel is maximum. The control law is based on a feedback controller, including as control parameters the lateral error, angular error and the curvature (see Figure 5.1b). The control matrix is defined as follows:

[2]http://www.civitec.com/

$$U(t) = \begin{bmatrix} k \\ L_{error} \\ A_{error} \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \tag{5.2}$$

where k is the curvature, L_{error} is the lateral error and A_{error} is the angular error and α_1, α_2 and α_3 are the controller gains. These errors are calculated with respect to the difference between the look-ahead control point (placed at the frontal bumper of the vehicle) and the reference path. The L_{error} has a proportional effect for the control action, since it is associated to the reference error in y. According to the control law, it is a Proportional Derivative (PD) controller, which offers a fast response to keep the vehicle in the center of the lane, improving path tracking.

5.1.2 Trajectory planning

This section describes the simulated tests carried out. First, the trajectory planning module validates the path planning capabilities when facing different road layout configurations and turns. Roundabouts are also included, being an special case and addressed as described in Section 3.5. Then, the trajectory was generated by adding to the path a real-time speed profile generation, increasing comfort and validating the proposed approach in simulated urban environments.

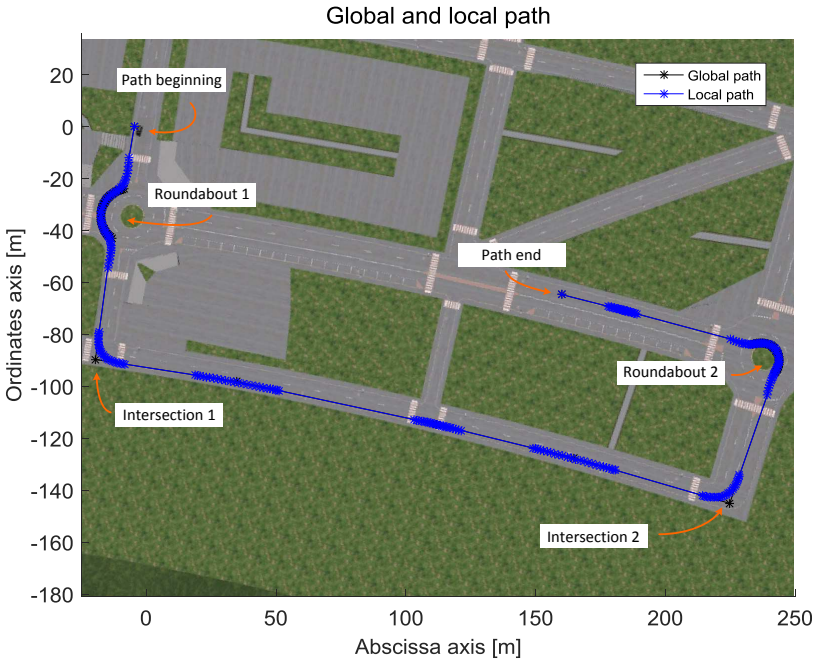
5.1.2a Path planning in straight and curved segments

The proposed path planning algorithm is able to manage intersections and roundabouts. Bézier curves have been implemented to create G^1 continuous paths and reduce curvature jumps. Figure 5.2a presents the path planned by the vehicle going through intersections and roundabouts in urban environments. The global path is depicted with a solid black line and the local smooth path is the blue line.

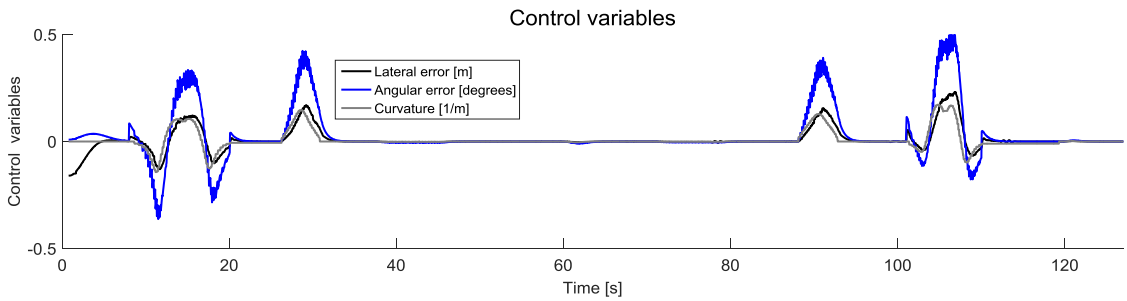
The simulated vehicle in Pro-SiVIC navigates the path with a maximum speed of 4m/s. Figure 5.2b presents the variables implemented by the control stage, taking into account the lateral error, angular error and the curvature of the path (see Equation 5.2). Four intersections where traversed by the vehicle: two intersections (at 30s and 93s) and two roundabouts (at 16s and 107s aproximately).

Figure 5.3 plots the two intersections. The chosen curve for the local path is shown in blue, the global path is shown in grey and the vehicle performance is shown in black. The trajectory of the vehicle is coherent with the path in blue without much lateral displacement for both intersections, i.e. Figure 5.3a and 5.3b.

Figures 5.3c and 5.3d present the curvature of the local path, showing the continuity throughout the curve and being lower than the maximum curvature feasible by the vehicle (shown in red). Finally, Figures 5.3e and 5.3f show the vehicle behavior, in detail, for each intersection. Both figures have a similar plot, where the lateral error is the solid black line, the angular error is plotted as the solid blue and the curvature in a solid grey line (all of them with respect to time). The sawtooth form of the variables is due to the discrete nature of the path, however, error values stay within correct limits, having a maximum lateral displacement of less than $0.2m$ in both cases. Please notice that the vehicle takes approximately two seconds to converge back to the path reference after the curve, presenting a stable behavior thanks to the low lateral displacement in the curved segment induced by the smooth behavior of the the planned curve.



(a) Local path generation in simulated urban environments



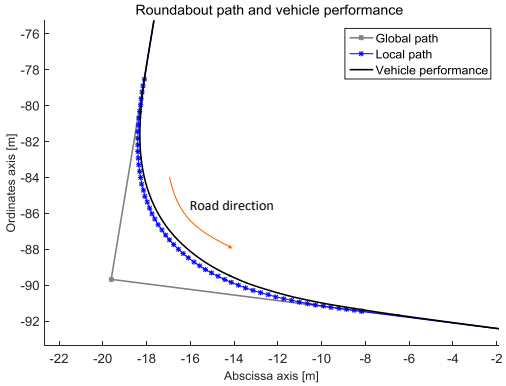
(b) Control variables from the whole path

Figure 5.2: Planning and control of the simulated vehicle

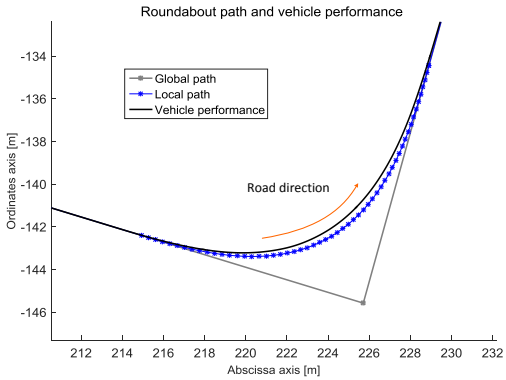
5.1.2b Path planning in roundabouts

Straight and curved stretches can be addressed by the trajectory planning module as shown so far. However, roundabouts have the particularity of multiple curvature changes, requiring a different approach to effectively plan an accurate path. This section shows the validation of the approach explained in Section 3.5.

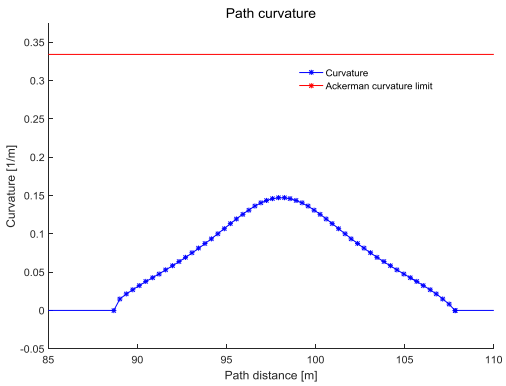
Roundabouts from the simulated path are as shown in Figure 5.4. Figure 5.4a depicts the first roundabout in the path. Inner and outer lanes are of $7.5m$ and $10.5m$ respectively, where the path chosen passes through the outer lane, following french traffic rules and the analysis made in Section 3.5.3. The global path is presented in grey, it describes the circulatory roadway from the entrance point up to the planned exit. The local path is the solid blue line, describing a smooth path that joints straight segments (entrance and exit) with the curved one (circulatory roadway). The vehicle performance is depicted with the solid black line, describing a trajectory that follows the pre-planned path, having small lateral displacement. Figure 5.4b presents the second roundabout taken in the experiment. The global path, the local path and the vehicle performance are depicted



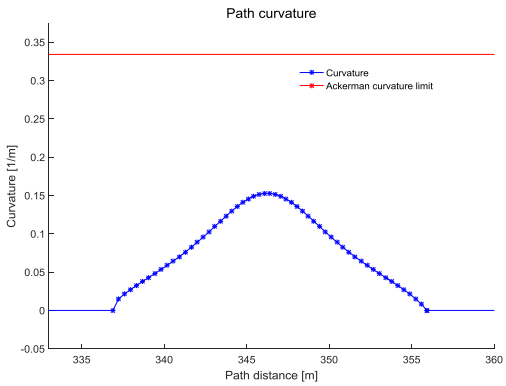
(a) Intersection one from Figure 5.2a



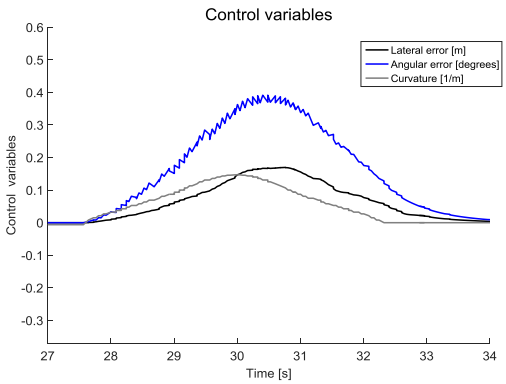
(b) Intersection two from Figure 5.2a



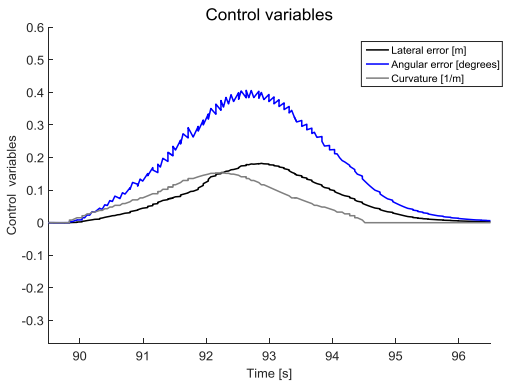
(c) Curvature of intersection 1.



(d) curvature of intersection 2.

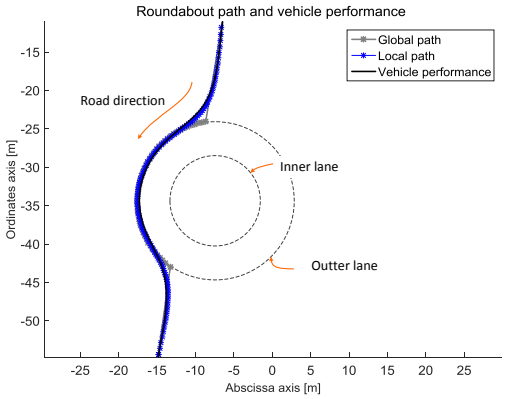


(e) Control variables at intersection 1.

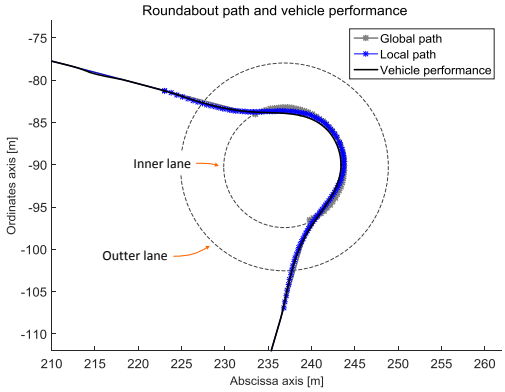


(f) Control variables at intersection 2.

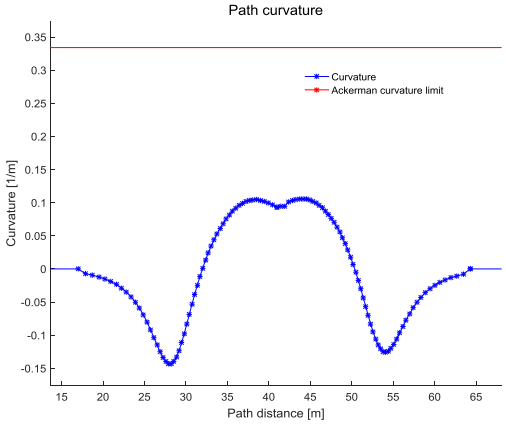
Figure 5.3: Path generation at turns.



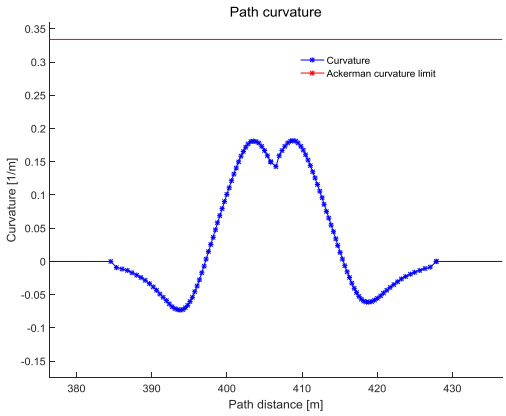
(a) Roundabout one from Figure 5.2a.



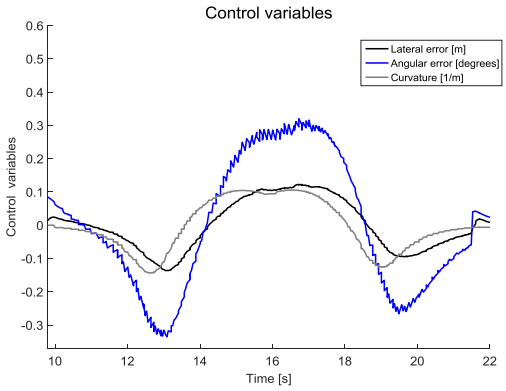
(b) Roundabout two from Figure 5.2a.



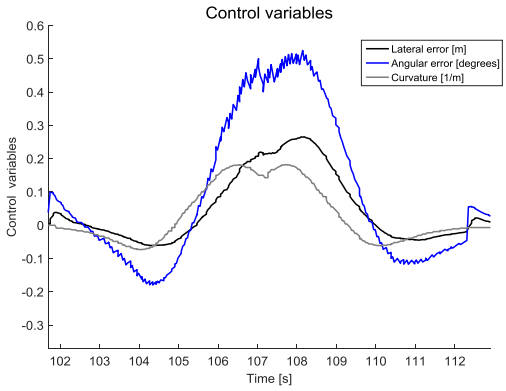
(c) Curvature of roundabout 1.



(d) curvature of roundabout 2.



(e) Control variables at roundabout 1.



(f) Control variables at roundabout 2.

Figure 5.4: Path generation at roundabouts.

in grey, blue and black respectively. The roundabout radii for each lane are $10m$ for the outer one and $7.5m$ for the inner lane. The trajectory described by the vehicle complies with the blue planned path. Also, the lane taken is the inner one since the next stretch is located further away than half of the roundabout perimeter (see Section 3.5.3).

The curvature profile for both roundabouts are depicted in Figures 5.3c and 5.3d. Although the local planning only assures G^1 continuity, it is visible that the curvature discontinuity between all roundabout phases is minimal. In Figure 5.3c the entry and exit curves are close to symmetrical. This also happens in Figure 5.3d, however, each curve is optimized in a different process and results are slightly different for entry and exit curves of the same roundabout (given the different entry and exit angles). Figures 5.4e and 5.4f present the variables taken by the controller at both roundabouts. The lateral displacement shows a good behavior, always under $0.25m$ for the whole experiment. The angular error (in blue) shows also a good behavior under $0.5rad$ through the experiment, where its sawtooth form comes from the discrete nature of the local path. Finally, the curvature is depicted in grey. Results in this section have been published in [Gonzalez et al., 2014].

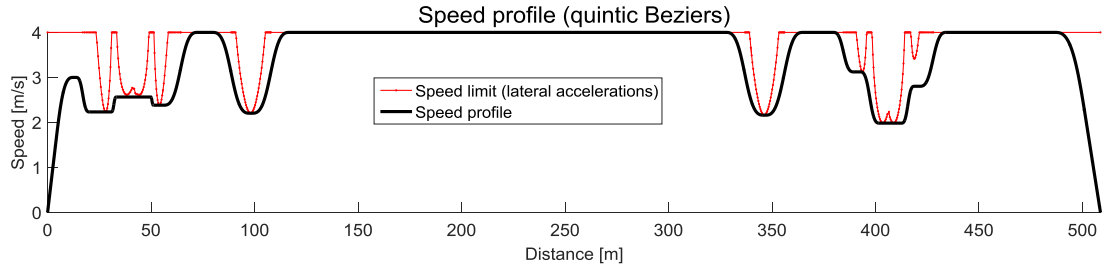
5.1.2c Speed profile generation

Once the path is planned, a speed profile is needed to indicate the vehicle the correct speed for each path segment. The present section shows the validation tests for the quintic Bézier speed planning method in simulated environments, considering the path and its curvature to increase comfort, as proposed in Section 3.4.3.

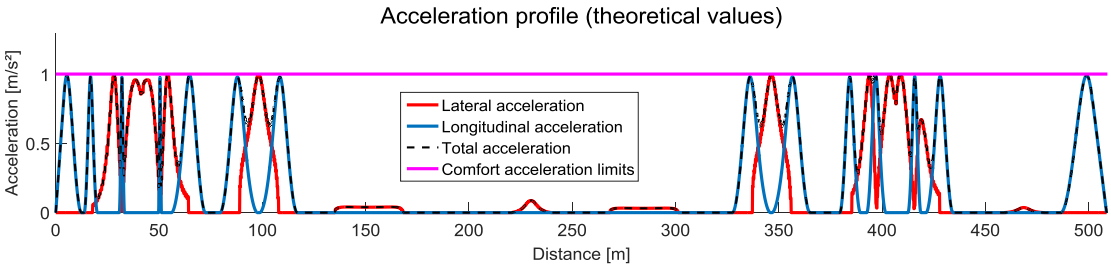
The advantages of the quintic Bézier method over the jerk limitation method where presented in Section 3.4.3c, i.e. the proposed method with Bézier curves ensures a smooth behavior, complying with acceleration limits and increasing comfort. Quintic Béziers have been therefore implemented to create a comfort speed profile over the path in Figure 5.2b. Figure 5.5a shows the speed profile computed with the quintic Bézier approach. The black line represents the speed profile with respect to the curvature profile of the geometric path in Figure 5.2b. The addition of both will create the trajectory to be followed by the vehicle. The maximum velocity is shown with the solid red line, which complies with lateral acceleration limits ($1m/s^2$ for the present experiment) and serves as a reference for the reader to evidence the effect of the curvature over maximum speeds (please see Equation 3.18).

Figure 5.5b presents the absolute theoretical values of lateral, longitudinal and total (euclidean) acceleration of the vehicle (see Equation 3.19). The longitudinal acceleration of the vehicle is as shown by the blue curve, while the red one depicts the lateral acceleration and the black dotted curve represents the euclidean sum of both red and blue profiles. The acceleration limit, set at $1m/s^2$, is never surpassed. Curves placed approximately at $150m$, $230m$ and $280m$ do not possess curvature values that oblige the vehicle to decrease the velocity, so the speed is kept as maximal.

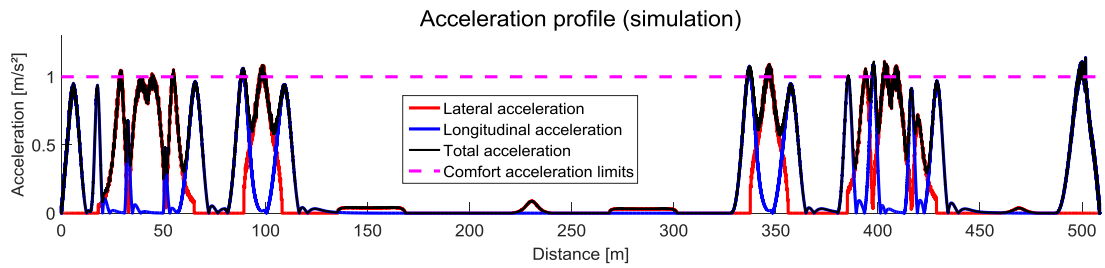
Vehicle's accelerations responses are presented in Figure 5.5c. The vehicle slightly surpasses the acceleration limit (magenta line at $1m/s^2$), because of the delay introduced by the vehicle model behavior. This affects the longitudinal acceleration profile in a way that accelerating and decelerating curves are within a maximum error range of 5%. This effect is linked to the way the vehicle follows the speed profile and also to Equation 3.22. When accelerating, this translates to a smaller maximum slope $M(t)$ and a lower than expected longitudinal acceleration peak (see the initial blue curve in Figure 5.5c). When decelerating, the opposite effect occurs, i.e. the theoretical estimation is surpassed (see the blue curve at the end of the path in Figure 5.5c). Also, the intrinsic delay in the vehicle model behavior changes the positioning of longitudinal speed changes, thus the speed variation finishes some meters later than foreseen, affecting and increasing



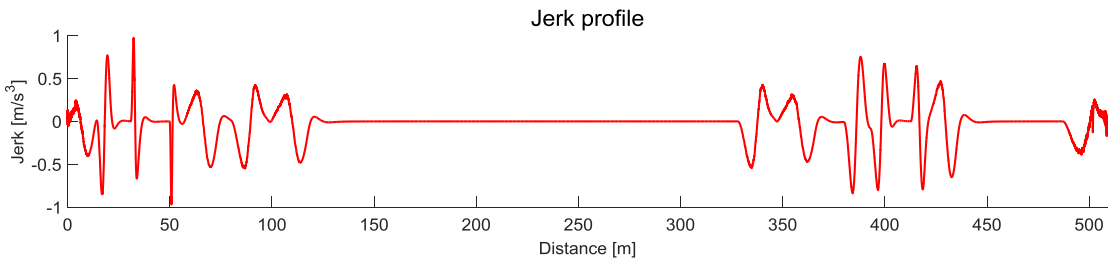
(a) Speed profile and curvature speed limit



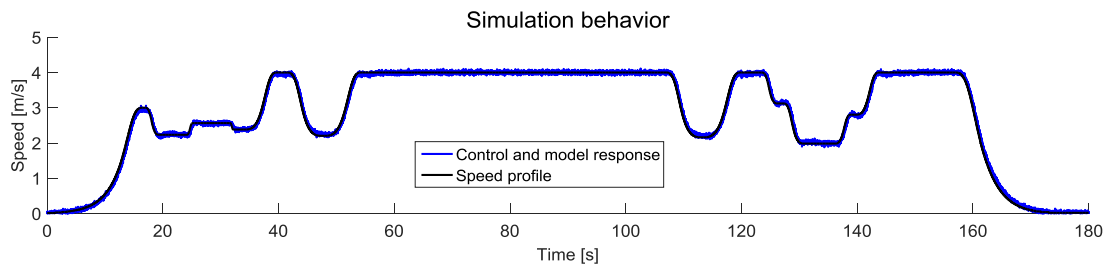
(b) Theoretical lateral, longitudinal and total accelerations



(c) Acceleration profiles from the simulation

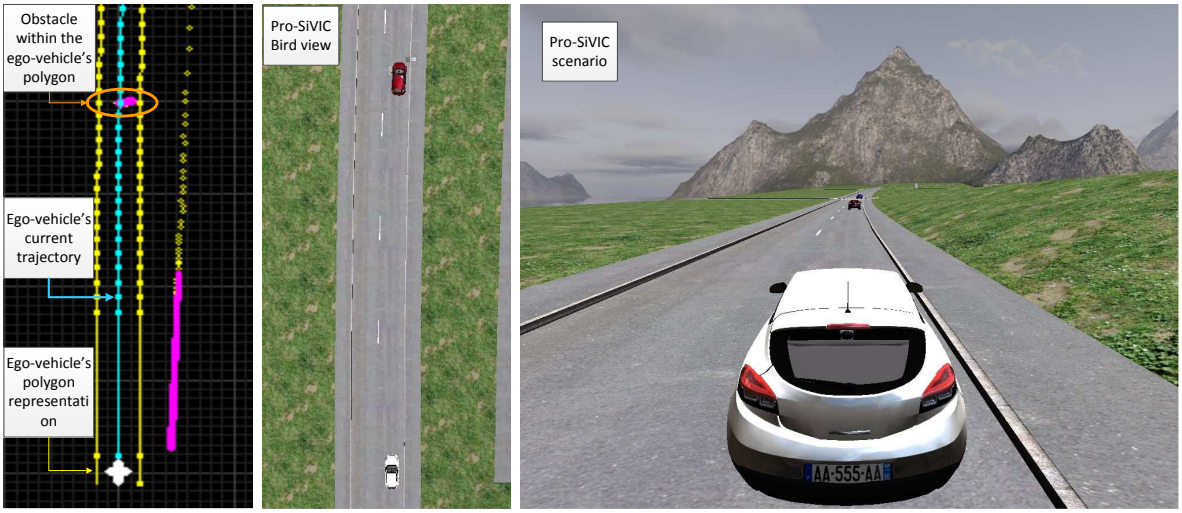


(d) Longitudinal jerk profile of the simulation



(e) Controller and vehicle model behavior

Figure 5.5: Proposed speed profile for the path shown in Figure 5.2b implementing the quintic Bézier method.



(a) (b) (c)

Figure 5.6: Representation of the obstacle detection and polygon evaluation in simulated environments. (a) shows the information coming from the perception stage, highlighting tracked obstacles in magenta; the polygon representation comes from the decision stage, and is represented in yellow, while the original path of the vehicle is highlighted in teal. (b) depicts a bird view of the the ego-vehicle in white and the first obstacle (the red car). (c) Camera view of the Pro-SiVIC scenario to test the functional architecture and avoidance maneuvers.

lateral accelerations (this behavior is normally present when decelerating, see Figure 5.5c at $340m$ and $400m$ for precise examples).

The measured jerk is presented in Figure 5.5d. It is always within $\pm 1m/s^3$ and keeps a smooth behavior throughout the planned speed profile. Finally, the vehicle model and control response are depicted in Figure 5.5e. The speed profile is shown in black, followed by the blue curve representing the behavior of the vehicle (Figure 5.5a differs from Figure 5.5e since the former depends on distance whereas the latter depends on time). The model response shows good results, following the speed profile and showing acceleration overshoots of only extra 5% as discussed above. Results of this section have been published in [González et al., 2016a].

5.1.3 Avoidance maneuvers

Once the trajectory planning was tested in simulation, including its ability to address most road layouts (i.e. intersections, roundabouts) with a smooth speed profile generation, the system capacity for negotiating obstacles was evaluated. This section describes the simulations carried out for validating the avoidance maneuver algorithm introduced in Chapter 4.

The obstacle avoidance component gives dynamicity to the local planning module in the decision stage and allows to avoid obstacles by implementing overtaking maneuvers. The information needed to plan new driving strategies to avoid obstacles comes from the perception stage which characterizes all obstacles in the field of view of the vehicle. Figure 5.6 shows a graphical example of the information received from the perception algorithms and the scenario for the obstacle avoidance tests. Here, the ego-vehicle is represented by the white vehicle, while the obstacle is represented by the red vehicle.

Figure 5.6a depicts the information from the perception stage, where obstacles are in magenta.

The current trajectory of the vehicle is shown in teal and the polygon predicting future ego-vehicle's positions is drawn in yellow. In this situation, the vehicle detected that the obstacle is within the polygon and therefore a new trajectory is needed. Figure 5.6b gives a bird view of the ego-vehicle and the first obstacle (a static one) in Pro-SiVIC. The same situation is represented in a camera view in Figure 5.6c, for a better understanding of the current scenario.

Figure 5.7a depicts a more complete view of the whole testing scenario, where static and dynamic obstacles are present. The ego vehicle is represented by a white rectangle and its borders color represent the temporal dimension. The experiment was done in the Pro-SiVIC simulator from $T = 0s$ to $T = 44.84s$, where the ego vehicle travels from "A" to "B" and overtakes two obstacles. The first one (depicted with a black border rectangle) is a static obstacle, overtaken by the ego vehicle at a constant speed of $7m/s$ implementing the three stages explained in Chapter 4. The second one (depicted by a rectangle with a black center) is a moving vehicle at $2m/s$. The temporal relationship between both vehicles is shown in colored borders of both, the ego-vehicle's rectangle and the obstacle's, where the overtaking maneuvers are employed.

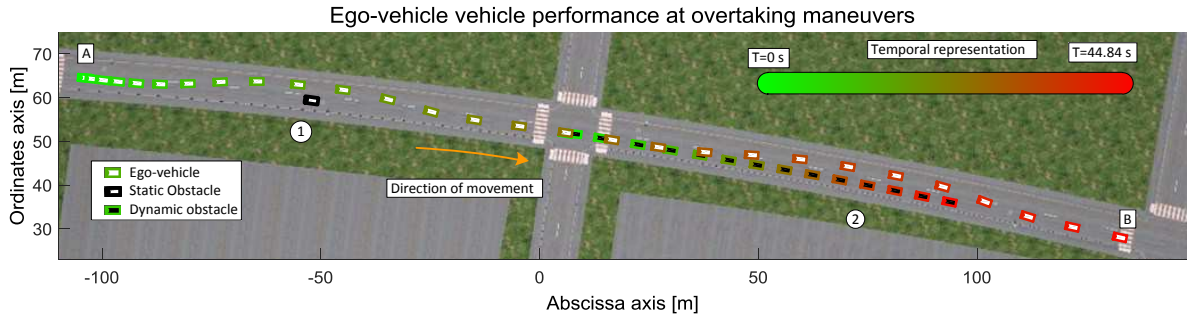
Figure 5.7b presents the path planed by the ego-vehicle. At first, the red path is given to the controller and this one follows the path reaching a speed of $7m/s$. Once the static obstacle is detected, the avoidance maneuver starts and plans the first curve to take the ego-vehicle into de adjacent left lane and overtake the static obstacle. The avoidance path is shown in blue, followed by a straight line (depicting the passing section) and finished by a path returning to the original lane. Once the returning path is identified, both paths are joined and the vehicle effectively engages once more the original path.

Quintic Bézier curves aid in joining both (red and blue paths) with G^2 continuity, while keeping comfort. Figure 5.7c shows the continuity of the curvature throughout the path navigated by the vehicle. Markers one, two and three in the figure indicate the curvature in the overtaking maneuver executed to pass the static obstacle. Markers five, six and seven present the curvature for the path taken while overtaking the moving obstacle, belonging to the avoidance, passing and returning phase respectively. The beginning of the path, marker four and the end of the path belong to the original trajectory (in red) and show no difference in continuity from the overtaking maneuver. The comfort is assured since the maximum curvature of the path (represented with a red dashed line) is kept under $0.204\left[\frac{1}{m}\right]$, keeping lateral accelerations under the comfort value of $1m/s$ for the ego-vehicle traveling at $7m/s$ (see Equation 3.18).

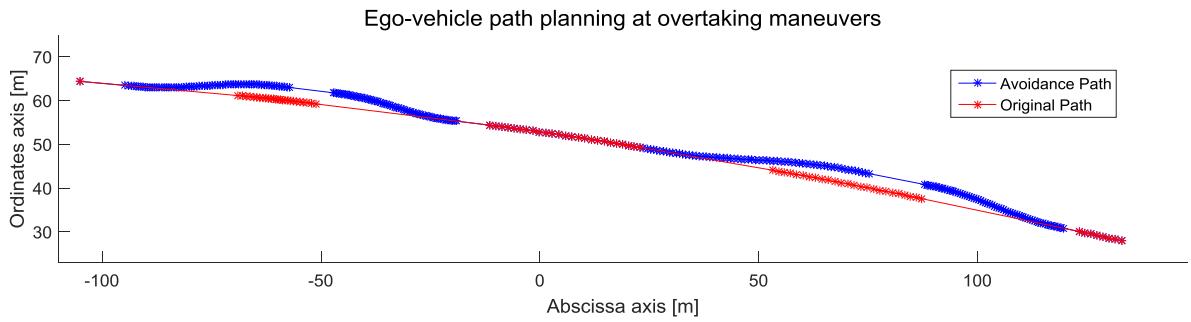
Figure 5.7d presents the control variables, with respect to time, for the lateral control of the ego-vehicle. The black line shows the lateral error while following the avoidance path. It remains within the interval [-0.1,0.1], keeping the vehicle close to the path and having a peak lateral deviation of $0.9m$ around $18s$. The angular error is depicted in blue, having a sawtooth behavior due to the discrete nature of the path. However, its values are low (its peak value reaches only up to $0.056rad$), showing a coherent behavior while navigating the path. The curvature is given in grey and resembles the theoretical one, keeping comfort and continuity throughout the path.

The functional architecture is able to create the needed trajectories in real-time. Figure 5.8 presents the temporal evaluation for the future collision detection and the whole obstacle avoidance component. Figure 5.8a shows the time consumed, in milliseconds, to verify that obstacles are or not within the current trajectory of the vehicle. This is done with respect to the original path if no avoidance maneuver is in course and with respect to the overtaking path if the avoidance has been triggered. The evaluation is very fast (about $0.8ms$ in worst case scenario) allowing a fast computation of the necessary path for the avoidance.

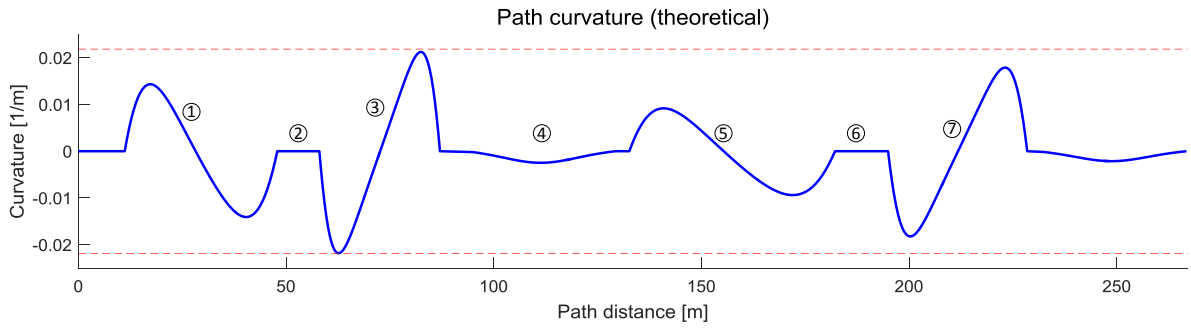
Figure 5.8b presents the time consumed while executing overtaking maneuvers. The time is expressed in milliseconds, showing a fast path generation with an elapsed time lower than $25ms$



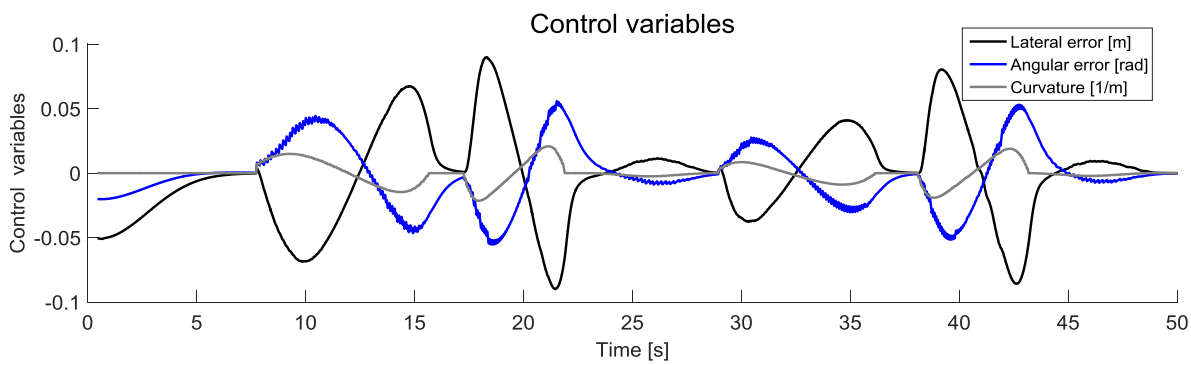
(a) Temporal representation of the ego-vehicle states (rectangle with white center), avoiding static obstacles (rectangle with black borders, see marker one) and dynamic ones (rectangle with black center, see marker two).



(b) Path planning for avoidance maneuvers.

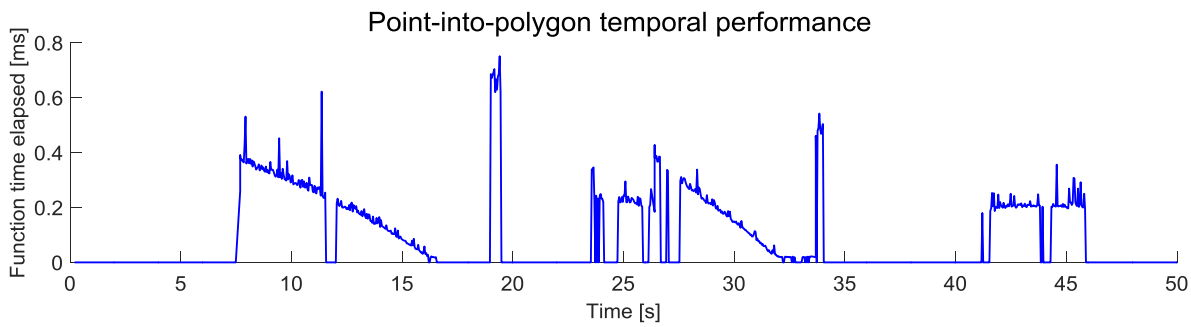


(c) Theoretical curvature for the avoidance path.

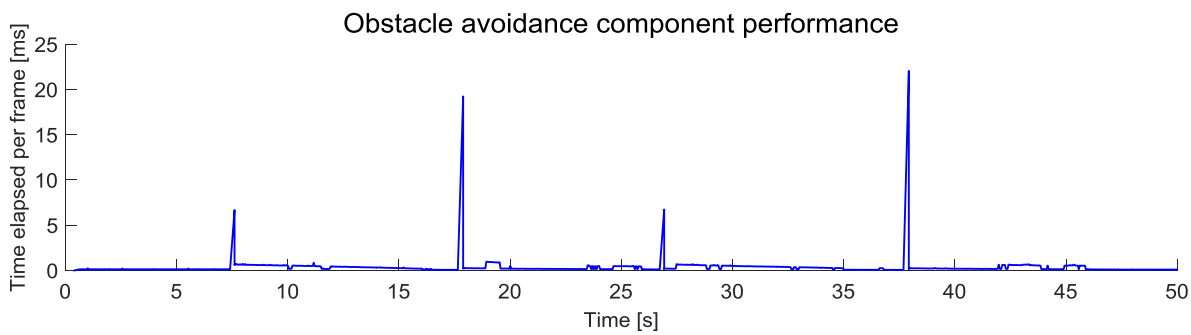


(d) Control variables from the ego-vehicle while following the avoidance path.

Figure 5.7: Obstacle avoidance in simulated urban environments.



(a) Performance of the future collision verification



(b) Performance of the obstacle avoidance component

Figure 5.8: Temporal performance evaluation of the future collision detection function and the whole obstacle avoidance component.

per path. The generation of a new path is done only as needed and not all the time, which decreases the time consumption of the component and allows for a fast collision detection when the component is not generating a new strategy. This is visible in Figure 5.8b at seconds seven and 26, where the planner is generating the first avoidance phase of the overtaking maneuver for the static and dynamic obstacle respectively. Seconds 17 and 37 show the time consumed by the component when returning to the original path. This last is more time consuming because the phase to return to the original path needs to seek the shortest path back, while keeping comfort constraints chosen by the driver/passengers as explained in Section 4.2.2c. The time consumed by the future collision detection is also visible here, with a temporal behavior under $1ms$.

5.2 Validation tests—real platforms

This section describes the experimental tests carried out. Starting from the simulation environment, the developed algorithms were translated and adapted to the INRIA experimental platform. Below, this platform is firstly introduced. Then, the different experiments to validate the developed architecture that were carried out in Simulation are done with the platform in real urban scenarios.

5.2.1 The Cyberbus

Real tests where performed using an electric minibus system called Cyberbus (or Cybus, see Figure 5.9). It is an electric minibus system without mechanical driver actuators. It is capable of traveling autonomously from one point to another in a road network. This vehicle has been designed at the

Figure 5.9: Cybus platform

INRIA-Rocquencourt by the RITS team (former IMARA). A three months demonstration showed that fully automated road vehicles are suitable for public transport [Bouraoui et al., 2011]. Other demonstrations have also implemented such system in some European project demonstrations, as in CATS and CityMobil2 [Roldao et al., 2015].

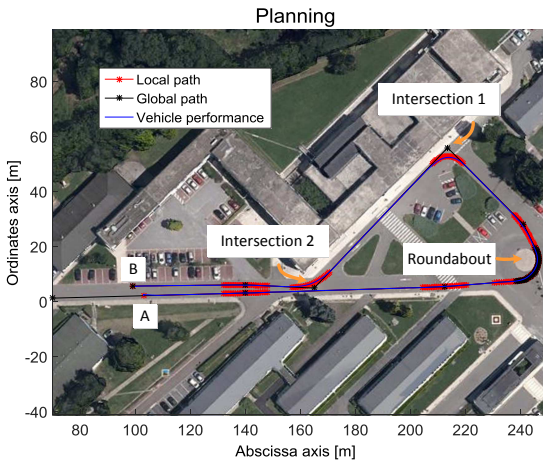
For localization, a SLAM algorithm is used. It is capable of estimating the joint posterior probability over all past observations using real time information for a large environment reconstruction, as explained in [Xie et al., 2010]. The environment data is collected by two Ibeo Lasers on the top front and rear parts of the Cyberbus (Figure 5.9). Obstacle detection is also possible with the two Sick lasers located on the front and rear down parts of the vehicle. These lasers are capable of detecting an obstacle within a 50m range, avoiding possible collisions. The acceleration is tracked by an IMU (located over the rear axle). Two incremental encoders are responsible of measuring the speed (attached to the engines in the rear axle) and a third one tracks the steering angle of the wheels (over the front axle).

The Cybus is equipped with two computers. The first is the master, it manages the steering, speed, obstacle detection, tracking and control. The second is the slave PC and it is dedicated to localization algorithms, receiving directly the information from the lasers and performing the SLAM. It sends the information to the master PC via Ethernet connection. Finally, a joystick and a touchscreen are part of the HMI. The vehicle is powered by a 72 Volts supply conformed by six Li-Ion batteries. For safety, five emergency stop buttons and a Remote Radio Unit (RRU) are capable to suddenly stop the vehicle in case of emergencies.

5.2.2 Trajectory planning

The test environment is presented in Figure 5.10a (as a bird view of the tracks used for the experiments) and Figure 5.10b (as a camera view of the roundabout in our research center). The test track used for the real experiments is a private urban environment within INRIA Rocquencourt. The main characteristics of the test track are:

- The vehicle travels from point *"A"* to point *"B"* depicted in Figure 5.10a facing two intersections and a roundabout.
- The radius of the roundabout is $8.5m$,comprising four entrances/exits.
- The roundabout exit taken by the vehicle is linked to a curved segment, where G^1 continuity is kept and the curvature jump minimized.



(a) INRIA facilities

(b) Roundabout and Cyberbus on real implementation.

Figure 5.10: Real platform and controller adjustments.

5.2.2a Path planning in straight and curved segments

To validate the generation of the path generation in real platforms, different turns and road layouts are present in the experiment. In Section 5.1.2a these where validated in simulations, showing good results. The present section shows the generation of the path in the Cybus.

Figure 5.11a presents the planning and vehicle behavior for the first intersection. The global path is represented in black, the local path in red and the vehicle behavior while following the path is in blue. The behavior of the control variables is as depicted in Figure 5.11c. The angular error is shown in blue and never surpassing the $0.4rad$, its sawtooth behavior is due to the discrete behavior of the path and the localization errors of the SLAM system (with a maximum error of $0.15m$). The curvature is shown in grey, never surpassing the curvature limit of the vehicle ($0.33\frac{1}{m}$). The lateral error is plotted with the solid black line and shows a good behavior for the vehicle, which never surpasses the $0.25m$.

Figure 5.11e presents the total accelerations associated to the embedded IMU. Longitudinal, lateral and vertical accelerations are shown in black, blue and magenta respectively. Notice that lateral accelerations only appear while the vehicle travels through the curved segment. Longitudinal accelerations are small, since real experiments where done at constant speeds of $8km/h$.

The second intersection is plotted in Figure 5.11b. The global path, the local planning and the vehicle behavior arer in black, red and blue respectively. The control variables behave in a more smooth way than at the first intersection. This is as expected due to the wider angle described by the second intersection. Figure 5.11d shows the control variables of the vehicle: the lateral error in black, the angular error in blue and the curvature of the path in grey. Notice that the lateral error keeps within an error of $0.2m$, the angular error is always less than $0.15rad$. The accelerations associated to the movement of the vehicle through the intersection are presented in Figure 5.11f. Here, the lateral acceleration (in blue) keeps below $0.15m/s^2$ keeping a comfortable profile for passengers.

5.2.2b Path planning in roundabouts

Roundabouts have also been included in real tests. Once validated in simulations (see Section 5.1.2b), real implementations in the Cyberbus require the tuning of the high-level controllers to acuratelly follow the path. Figure 5.12b shows the final tuning of the controller for the real roundabout. Several turns were performed at difference speeds (between 1.0 and 4.0 m/s). The maximum error is around $0.15m$, which is low, compared to the SLAM error [Xie et al., 2010]. Regarding the comfort while driving inside the roundabout, there is not oscillation, and due to the speed range used in the experiment, the lateral acceleration never exceeds $1.0m/s^2$—Fairy uncomfortable-based on the ISO 2631-1 Standard [ISO, 1997].

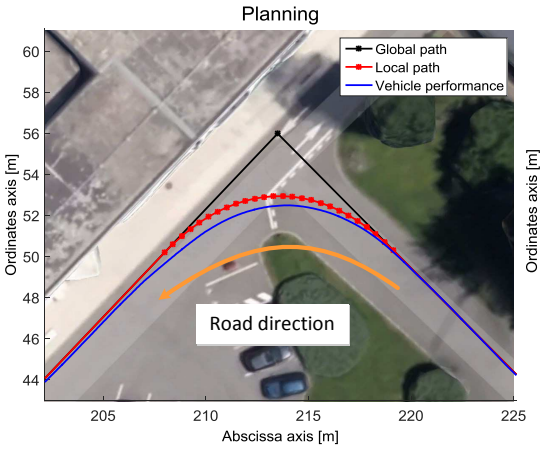
This experiment was carried at $8km/h$, which is a normal velocity for CTS platforms (see [González and Perez, 2013]). It shows how the vehicle (black line) follows the reference generated by the local path planning (in red). Figure 5.12c shows the behavior of the control variables, displaying the lateral and angular error and the curvature for the whole experiment. The lateral error is low and it keeps under a low value of $0.25m$, see Figure 5.12d, where the maximum lateral displacement of the vehicle is presented (inside the roundabout). Figure 5.12e shows the evolution of the accelerations read from the on-board IMU for the whole experiment. The continuous blue line shows the lateral acceleration, which increases when the vehicle enters the roundabout (see Figure 5.12f), but always keeping a comfortable driving (accelerations under $0.35m/s^2$ [ISO, 1997]).

Figure 5.12a shows the real-time roundabout path generation and depicts the three stages: entrance, circulatory roadway and exit. The entrance stage differs from simulations in that the entry angle is tangent rather than perpendicular to the circulatory roadway. This angle information is taken from the digital global map. In Figure 5.12d the curvature (in grey) grows up to $1/r$ (from 113 to 118 seconds approximately), where r is the radius of the roundabout ($8.5m$ in our case). The circulatory roadway keeps the curvature constant. The exit stage shows how the generation of the curve can handle curved exit paths. The curvature of the exit stage is depicted between seconds 123 to 130 of Figure 5.12d. Please notice that it reaches the curvature value of the next curved segment and a G^1 behavior depicted at the curve joint in Figure 5.12a. Notice also that the joint occurs around $130s$ of the experiment and no discontinuities are shown in Figure 5.12d around this time in the angular or lateral errors, evidencing the G^1 continuity and increasing comfort. The curve generation time for each intersection is around $2ms$ (milliseconds) in a core duo Intel system ($2.0GHz$). The presented results have been published in [González et al., 2017].

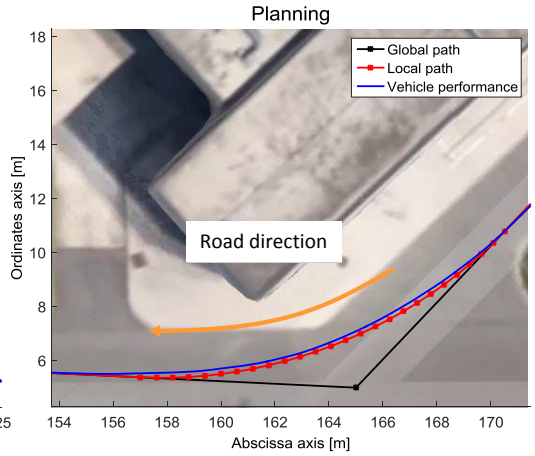
5.2.3 Avoidance maneuvers

In the previous sections, the functional architecture was able to take the vehicle through different intersections and road layouts, with high comfort up to its destination. This section presents the obstacle avoidance test in real platforms. Figure 5.13a shows the setup for avoidance tests implementing the Cybus as the ego-vehicle. Two experiments where carried out, with static and moving obstacles. In the first experiment, the obstacle is a red commercial Citroen C3, that simulates a vehicle breakdown in an urban environment. In the second experiment, first an stopped vehicle is the obstacle and then a moving Cycab (small electric vehicle) is encountered by the ego-vehicle.

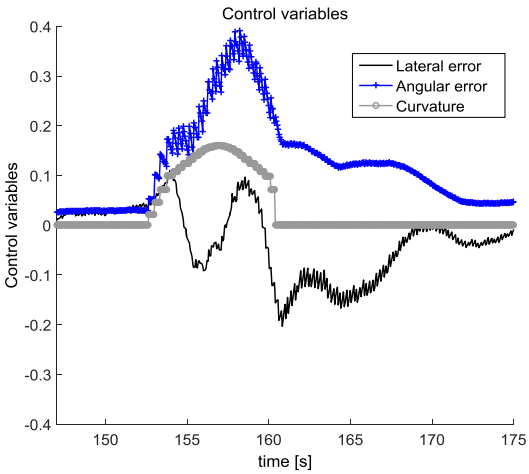
The Cybus is able to detect obstacles through laser sensors located in the lower front and back part of the vehicle (see Figure 5.9). Once the Cybus begins the same route shown in Figure 5.10a, the red car is detected as an stopped obstacle and the overtaking maneuver is initiated. Figure 5.13 shows the first experiment at different temporal captures, evidencing the proper overtaking of the stopped obstacle.



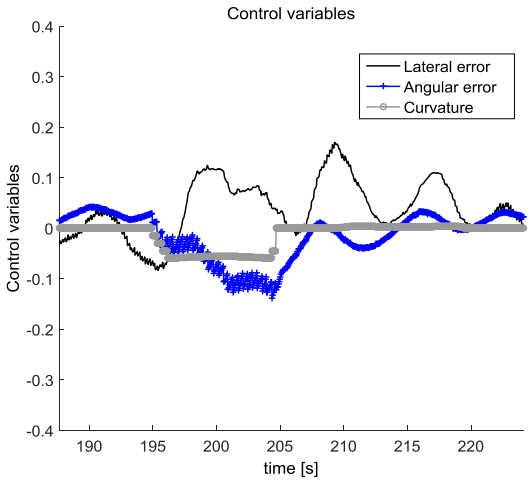
(a) Intersection one from Figure 5.10a



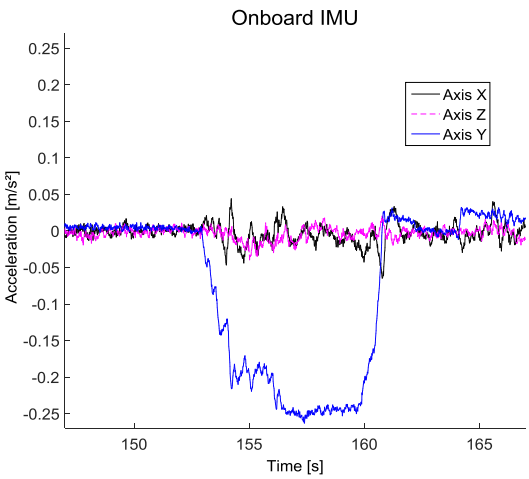
(b) Intersection two from Figure 5.10a



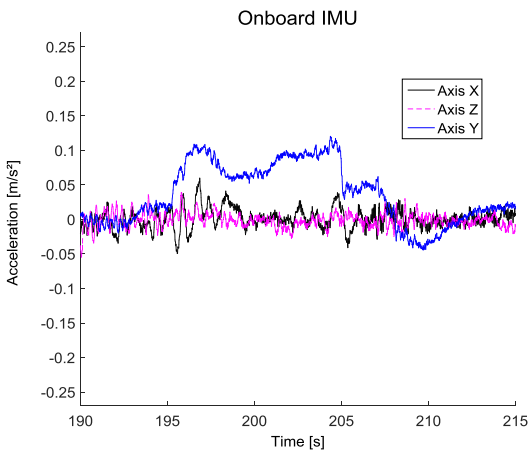
(c) Lateral control variables at intersection one



(d) Lateral control variables at intersection two

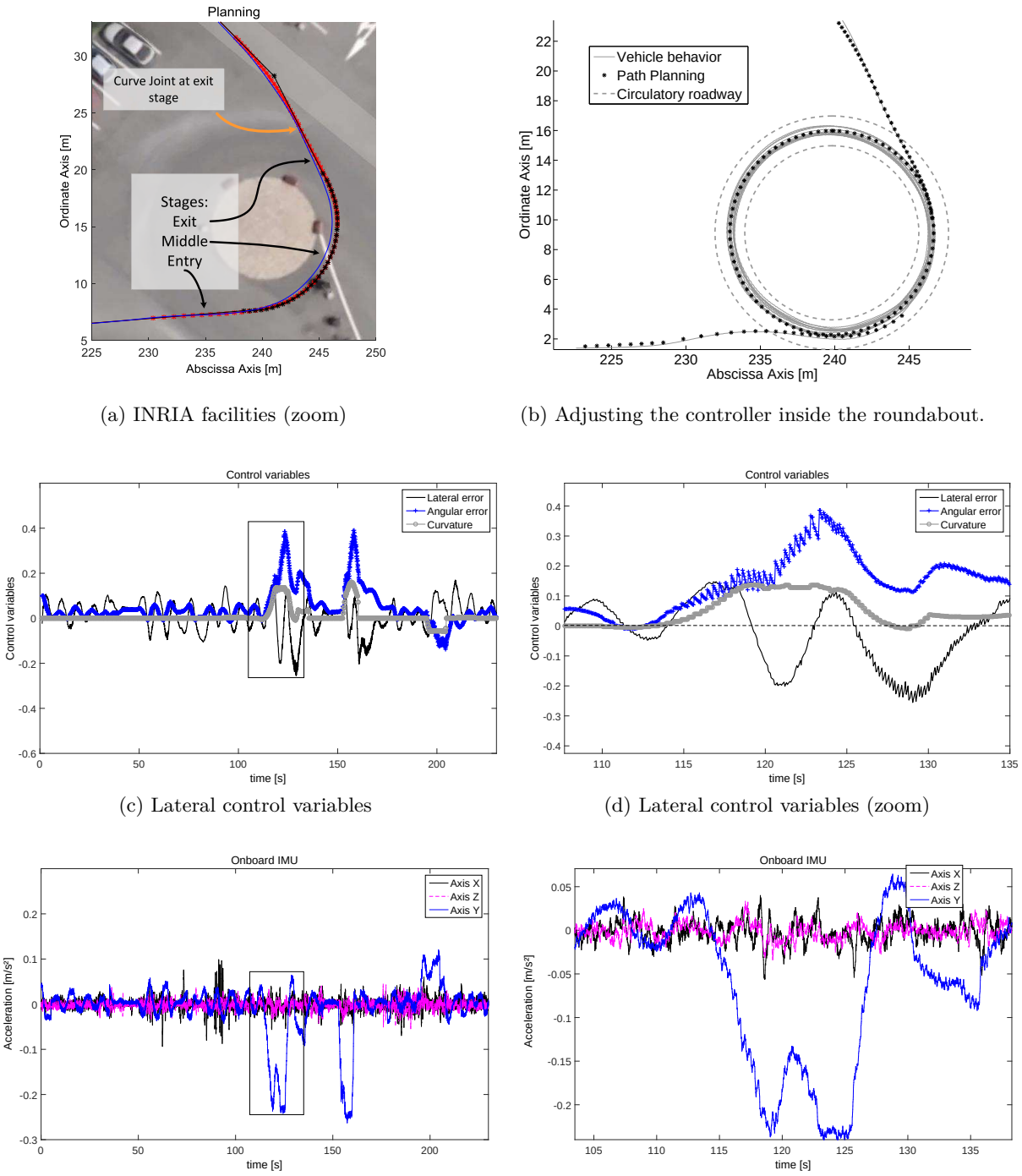


(e) On-board IMU recordings at intersection one



(f) On-board IMU recordings at intersection two

Figure 5.11: Real urban scenario: intersections.



(a) INRIA facilities (zoom)

(b) Adjusting the controller inside the roundabout.

(c) Lateral control variables

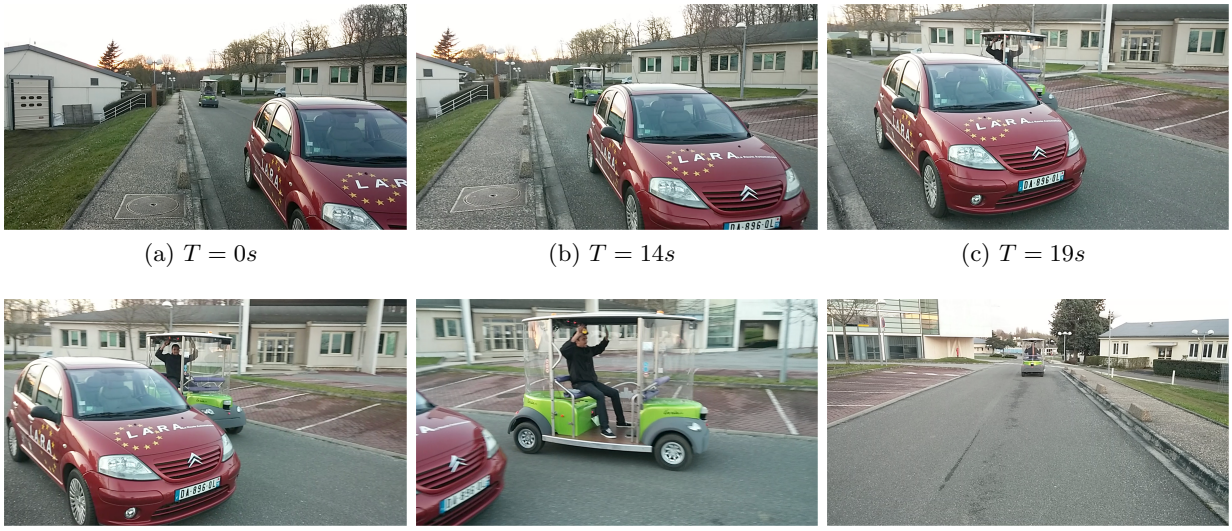
(d) Lateral control variables (zoom)

(e) On-board IMU

(f) On-board IMU (zoom)

Figure 5.12: Real urban scenario: the roundabouts.

The Cybus takes the information coming from the perception stage and compares the obstacles to its predicted future states, verifying if a future collision is detected. The red car is over the original path the Cybus intends to take. The point-into-polygon algorithm detects the future



(a) $T = 0s$ (b) $T = 14s$ (c) $T = 19s$

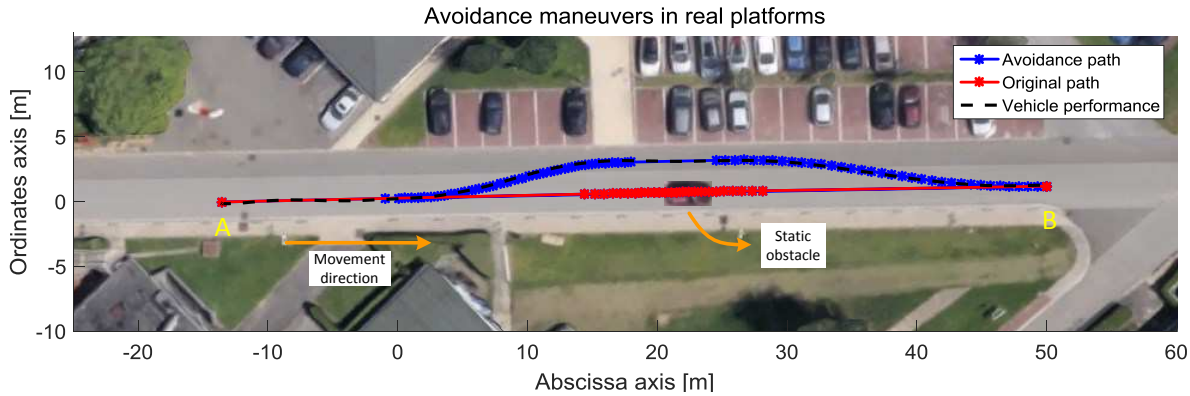
(d) $T = 21s$ (e) $T = 23s$ (f) $T = 33s$

Figure 5.13: Obstacle avoidance with the cybus.

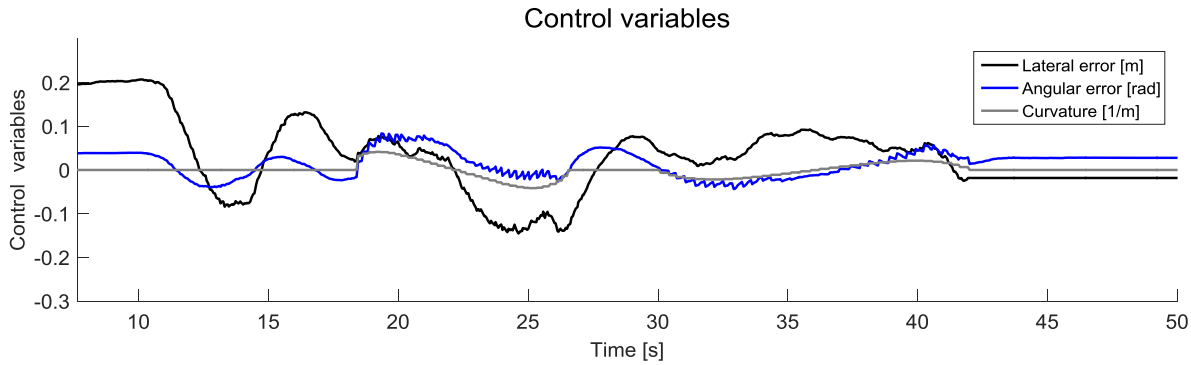
collision and, to avoid it, the overtaking is triggered. The path for the obstacle avoidance maneuvers is shown in Figure 5.14a, where the original path is in red and the avoidance one in blue. The ego-vehicle goes from "A" to "B" at $3m/s$, encountering a static obstacle in the middle of the path. Once the obstacle is detected, the overtaking maneuver is implemented to safely take the ego-vehicle to the left lane, pass the obstacle and then return to the original path. The ego-vehicle follows the planned red path until the avoidance maneuver is triggered, then follows the blue path and finally returns to the original lane as described by the black dotted line in Figure 5.14a.

Figure 5.14b depicts the behavior of the control variables. The lateral deviation is depicted with the solid black line, showing a good behavior and keeping values under $0.2m$ throughout the maneuver. The angular error in blue also keeps low values under $0.1rad$ and behaves in a sawtooth form when the path is curved, due to the discrete nature of the path. The curvature is presented in grey. It keeps a G^2 continuous profile (curvature continuous) with the original straight path. This is specially visible in curve joints with straight segments. Figure 5.14c gives the accelerations registered by the on-board IMU. The longitudinal acceleration is depicted in black, describing an initial acceleration at $10s$ and the braking action to stop the vehicle around second 43 (higher than the initial one due to the beginning of a slope), both under $2m/s$. The blue line relates to the lateral acceleration, having a low profile thanks to the smoothness and low curvature profile that characterize the path (the vehicle never presents a lateral acceleration over $0.3m/s^2$ and performs the overtaking maneuver as comfortable as possible).

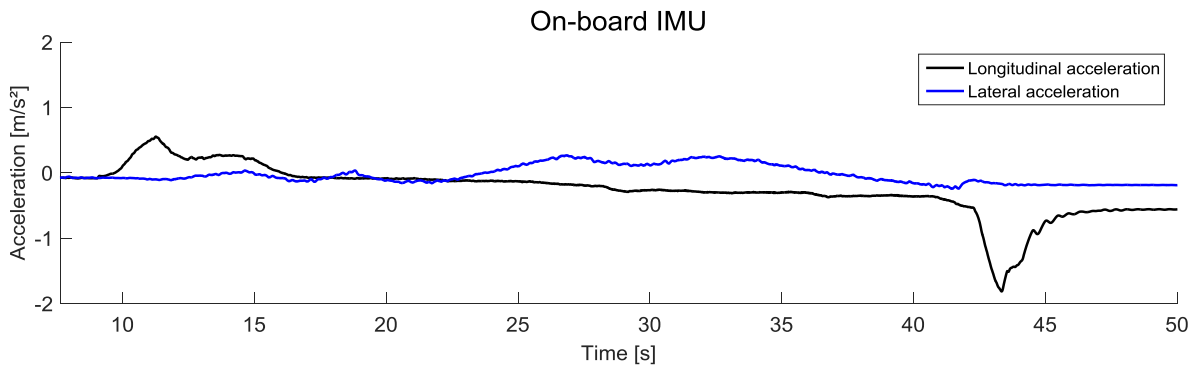
The second experiment was performed in the same area of the testing grounds. This is visible in Figure 5.15a, where the trajectories of the ego vehicle are visible, as well as the performance when following the path. The red dotted line in the figure depicts the original path the Cybus intends to take. The dotted blue line represents the first avoidance path where the first vehicle (a commercial Peugeot 206). Once detected, the Cybus initiates its first avoidance maneuver and goes to the left lane. Once the obstacle is overtaken, the Cybus intends to return to the original red path. However a Cycab (electric Cybercar) is found to be in conflict with the trajectory. It is moving at a constant speed of $0.5m/s$. The Cybus, moving at $2m/s$ initiates then a new avoidance maneuver even before finishing the returning phase of the first overtaking, thus facilitating the



(a) Original and obstacle avoidance path at INRIA facilities



(b) Control variables for the avoidance performed in the cybus



(c) Accelerations from the onboard IMU

Figure 5.14: Obstacle avoidance maneuvers in real platforms.

avoidance of the second obstacle. The path of this new avoidance is shown with a dotted green line in Figure 5.15a. It starts from the middle of the returning maneuver of the first overtaking (second 35 approximately), taking the vehicle to the left lane and back, effectively avoiding the moving obstacle. The behavior of the vehicle while overtaking the obstacles is shown with a solid black line, first following the red path, then the blue one and finally the green one.

Figure 5.15b presents the performance of the vehicle with respect to the control variables while following the real-time generated path. The lateral error is depicted with a solid black line, evidencing the correct path tracking of the Cybus through the experiment, being the maximum lateral deviation of less than $0.25m$. The angular error also presents a coherent behavior and its

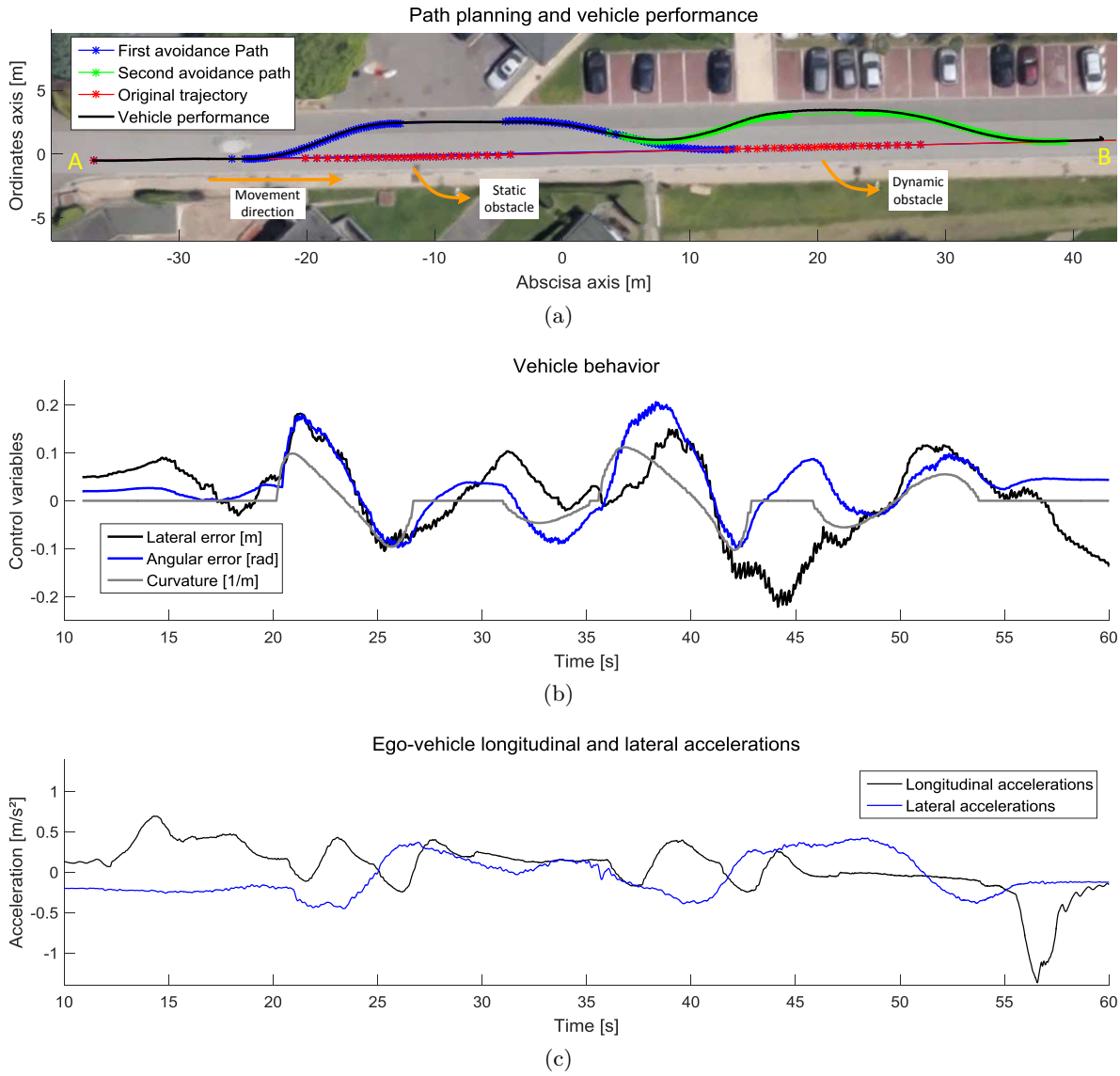


Figure 5.15: Consecutive obstacle avoidance maneuvers in real platforms.

value keeps under $0.2rad$. The solid grey line is the curvature. It is G^2 continuous throughout the whole experiment, increasing comfort and avoiding jumps in the path.

Finally, the longitudinal and lateral accelerations are shown in Figure 5.15c. Is visible that overall accelerations keep under a value of $1m/s^2$ showing a good comfortable behavior for passengers. However, between $55s$ and $60s$, the longitudinal profile shows a deceleration peak. This is due to the deceleration process of the vehicle, which happens in a terrain with a slight slope.

5.3 Shared Control and arbitration applications

Up to now, the proposed functional architecture for automated vehicles motion planning has been oriented to high automation applications. However, several challenges arise from automation, specially with respect to automated vehicles, where the social barrier (technology acceptance),

the legal gap and the technological developments are the main milestones yet to achieve [Van Schijndel-de Nooij et al., 2011], [Anderson et al., 2014].

Social acceptance comes from the confidence in the technology. The presented approach for motion planning has addressed partially this issue by including comfort in the design of the functional architecture, which is directly linked to confidence (the more comfort a passenger feels, the more acceptance he/she has towards the system). But to achieve full social acceptance, is clear that the system needs to be highly reliable and be able to give back the control to the driver when it fails (with sufficient time and information for the driver to take over the situation). Moreover, the lack of legislation with respect to the technology leads to a legal bottleneck, where the driver remains responsible for all that happens with the vehicle, even if the automated system intervenes. This is because in case of accident, crash responsibility remains unclear and is unknown how this will affect the industry or insurance companies. Also, the technological developments are related with the lack of standardization and the need for more accurate sensors and faster computing units to achieve real-time systems.

Issues in the acceptance of automated vehicle's technology could arise from human-automation interaction, specifically from a misunderstanding between the decision makers in joint actions—as in current lane keeping assistance systems (LKAS), when the driver wants to make a choice different than the system's corrective action [Abbink et al., 2012]. Since automated vehicles are still away from complete automation, the cooperation between the driver and the system is the clear next step. This allows to add the system's supervision over drivers' skills, supporting rather than replacing the human driver. Recent research points to this direction—e.g. [Baltzer et al., 2014], [Broggi et al., 2013a] and [Li et al., 2012]—arguing that this will permit to arbitrate and avoid conflicts between the decision makers [Baltzer et al., 2014], use the human driver as ground truth to thoroughly test the embedded system [Broggi et al., 2013a] and keep supervision over the driving task increasing drivers' and passengers' safety and comfort [Li et al., 2012].

The main challenge in developing human-machine systems which allow the cooperation of both decision-makers, is to arbitrate between the human and the automated system. This is since the balance between human adaptability and the precision of the embedded system is difficult to find, considering that manual control is prone to human error and highly automated tasks have still important limitations [Saleh et al., 2013]. Recent implementation of such systems propose stability envelopes as in [Erlien et al., 2016], where the arbitration acts as an "on-off" system, going from manual control to full automation and back, acting over the steering wheel. Other applications aim to smoothly shift authority with respect to LKAS applications [Saleh et al., 2013] and also introducing human machine cooperation examples or metaphors as in [Flemisch et al., 2014] and [Da Lio et al., 2015]. Moreover, the question of how the action suggestion is going to be provided has already been posed in [Li et al., 2012], where recent research points in the direction of applying the aid "as needed" rather than "all the time" [Flemisch et al., 2008a], [Petermeijer et al., 2015], thus avoiding drivers' over-reliance on automation.

The operation of the control sharing ADAS proposed in the present Ph.D. work is described in two main steps:

- **Risk assessment:** The first step is to calculate the risk of the situation in hand. To do this, the TTC and the Time To Limits Crossing (TTLC) are defined as the time to collision to the closest obstacle and the time to surpass the driving boundaries respectively. This permits to assess the performance of the driver. Fuzzy logic is then implemented to assess the risk of the current maneuver with respect to TTC and TTLC.

- **Shared control:** The second step is to arbitrate the level of authority of each decision-

maker by considering the risk of the current situation. This is done through haptic signals over the steering wheel and the strength of this signal is proportional to the level of authority granted to the automated system. The higher the risk, the higher the haptic help given by the system.

In the following, a review of these kind of systems is provided, as well as the development of the system, its addition to the base architecture and finally its validation in simulated environments.

5.3.1 Literature review

In the ITS field, shared control is the action of carrying a task simultaneously between the embedded system and a human driver, differing from manual control and full automation. The main challenge in this kind of systems lies in the decision making process and the assignment of control responsibility [Parasuraman et al., 2000].

When considering the driver in the control loop, the limits of responsibility for each decision maker are set up with respect to the automation level embedded in the vehicle. This permits the arbitration system to know the limits of the automation and, thus, the maximum authority of the system. The first levels of automation were set by [Sheridan and Verplank, 1978]. Here, ten different levels describe the amount of responsibility for each decision maker. [Flemisch et al., 2008a] presents a more developed view of the levels needed for control sharing, where the automation is based in the H-metaphor and clarified in two main groups: Tight rein and loose rein. In January 2014, a taxonomy of automated driving was issued by SAE International [SAE, 2014]; its control levels are depicted by Figure 2.1. It is based on the BASt and NHTSA previous standards but describing in more detail the three higher levels of automation.

The Arbitration concept is the process of settling an argument or a disagreement by an entity that is not involved. Although research in human-automation interaction exists since more than 40 years ago [Abbink et al., 2012], arbitration for automated vehicles has not been deeply explored. First approaches define cognitive states and relation between humans and machines [Hoc, 2001], also mental models as in human relationships have been considered by [Flemisch et al., 2008b]. This consideration leads to a scenario where the status of the driver and the system must be known, at all times, aiming to set an accurate level of automation for the current situation. The communication between the system and the driver should constantly occur, in a way that is possible for both to make mental models of each other [Flemisch et al., 2008b]. Also different metaphors have been stated, such as the copilot metaphor (or co-driver as in [Da Lio et al., 2015]) and the H-metaphor as a comparison between horse-human cooperation and vehicle-human cooperation [Flemisch et al., 2003].

5.3.1a Haptic systems

From other domains, where automation is already widely used (e.g. aviation, central rooms), it is known that automation has both positive and negative effects on the human operator. Thanks to this, increasing need to pay more attention to the human driver in interaction with the vehicle has been recently identified [Van Schijndel-de Nooij et al., 2011]. With increasing automation in the vehicle, these effects need to get far more attention on the short term, evaluating the human-vehicle relationship and assigning countermeasures if necessary. In order to have a regular communication between the two decision makers (the driver and the embedded system), in [Flemisch et al., 2003], a haptic HMI is proposed where active force feedback is the common language. This allows the message to be directly linked to the actuator where the reaction of the driver is expected, also

allowing the system to evaluate the performance of the driver. The haptic feedback can also give hints in terms of the action the driver should perform (e.g. the steering wheel turns a little to the right or left in order to hint the driver).

Haptic systems have been implemented widely across the literature: in gas pedal feedback [Abbink, 2006], [de Winter et al., 2008], and in steering wheel feedback [Mulder et al., 2008], [Saleh et al., 2013]. These are also used in learning simulators, improving the performance of drivers in different scenarios, even when sharing the driving task with a secondary one [Griffiths and Gillespie, 2005]. However, the use of corrective feedback is known to cause over-corrective behavior [Van Leeuwen et al., 2011] or bad performance when removed [Flemisch et al., 2010]. This happens because it impairs the input-output relationship in motor skill learning of the driver [Schmidt and Wulf, 1997]. In [Marchal-Crespo et al., 2010], the haptic aid shows a good performance if the feedback is provided as needed and not all the time. The HAVEit project [Flemisch et al., 2010], [Vanholme et al., 2013], and the InteractIVe project [Hesse et al., 2011] have made the first approaches into control sharing strategies, theoretically and in simulations, with driver-in-the-loop capabilities. Moreover, the DESERVE project included arbitration and control sharing as part of its rapid ADAS prototyping approach [Kutila et al., 2014]. The aim of arbitration and control solutions in ADAS, inside the DESERVE project, is to effectively share the control with the driver—stablish haptic signals as a communication bridge—and manage risky situations when the driver is inattentive or even fatigued. Part of the contributions of the this thesis have been developed in the framework of the DESERVE project.

5.3.1b Driver state assessment

In [Owsley and McGwin Jr., 2010], the importance of vision at the driving task was stated. Although visual acuity proved to be important, other indicators of the driver ability (e.g. the visual field, processing speed, divided attention, among others) have evidence for their relevance to the driver ability and safety, and can be measured in a non-invasive way with recent in-car perception systems, as in [Boverie et al., 2011]. Also, for arbitration and shared control applications, a state of the driver is needed in order to know his current status to perform the driving task. In [Zhang, 2011], an extensive study on driver distraction was performed. It showed that in terms of visual and cognitive attention sharing, while performing following or passing driving maneuvers, a warning from the HMI proved to be helpful.

Driver's limitations are related to his physiological and psychological states. An optimum pilot state includes an optimum alertness level and a task-oriented atention. The distinction between *alertness* and *attention* is justified in the way that driver *alertness* is presumed to be necessary but not sufficient for an appropriate focus on external events. Thus, drivers may be alert but still be inattentive. In order to assess alertness and atention two main factors are evaluated:

- **Drowsiness/fatigue:** Up to now, a universally valid definition of drowsiness still lacks. A tired driver mainly derives from performing a highly demanding task for extensive time periods ("time-on task" for the driving effort). Other definitions focus on the sleepiness level, which is the state of being ready to fall asleep. It is mainly caused by circadian rhythms and sleep disorders (reduced quality or quantity of sleep). [Sahayadhas et al., 2012] classifies drowsiness measurement methods in: 1) Vehicle-based measures: Assess driver's performance and gives a drowsiness probability with respect to deviation from lane center, acceleration/deceleration profiles, among others. 2)Behavioral measures: Assess driver's behavior, normally from in-vehicle sensors to detect yawning, eye closure/blinking, head pose, etc. from which drowsiness is detected; and 3)Physiological measures: Physiological

signals have a direct correlation with driver drowsiness—e.g. electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG), among others.

- **Driver distraction/inattention:** "Driver distraction refers to those instances when a driver's attention is diverted from the primary task of driving the vehicle in a way that compromises safe driving performance", [Young et al., 2008]. This distraction can be either internal (e.g. other passengers interaction, cellphone, etc.) or external (e.g. other road users, traffic signs, etc.). It can also be classified in different modes as: Visual (external attractors for example advertisement on the side of the road or internal attractors, e.g. looking to his children at the back of the vehicle, displaying an address onto a navigation device, etc.), acoustic (ringing phone, listening music) or cognitive distraction (conversing at phone, but also internal thought and rumination, etc.). A way to measure this is to detect the time that the driver is not looking at the road or even the time the driver shares its visual attention to a secondary task, as explained by [Boverie et al., 2011].

5.3.1c Legal and liability aspect

For automated vehicles, it is still unclear how legal and liability aspects are going to evolve. As a matter of fact, the U.S. legislation does not prohibit nor allows the use of automation in the driving task [Smith, 2012]. This leaves an important legal gap towards the responsibility of any action taken by the on-board system, since it is now an entity that "thinks for itself". Similar situations arise in Europe where in a crash the responsible at all times is the driver, even if an embedded system was controlling the vehicle [Trimble et al., 2014].

From the legal perspective, several initiatives in the U.S.[3] have already established some of the minimum safety requirements in order to allow automated vehicles technology [Anderson et al., 2014]. In the E.U., initiatives launched between governments and manufactures are currently creating the framework for the new standards and regulations for automated driving. These address legal matters and promote the standardization of the automated vehicles technology (e.g. FP7 Citymobil2 project [van Dijke and van Schijndel, 2012] and ECSEL-JU DESERVE project [Kutila et al., 2014]).

As to liability, [Beiker, 2012] noted that the situation is more complex with automated vehicles, concluding that it is unclear how the courts, or the public, will respond to the prospect of artificial intelligence acting on behalf of humans with life or death consequences. They expect that a set of policies can be established to create the necessary legal framework for further development of vehicle automation. In the E.U., the legal framework sets the liability of any crash towards the driver., raising many barriers for automated vehicles and restricting them to private roads. A helpful model of automation is to consider different levels of assistance and automation that can e.g. be organized on a scale as in [SAE, 2014]. This not only suggests but encourages the use of systems that consider the driver-in-the-loop. These will allow the industry to addition driver's vigilance to their systems supervision and avoid gaps (at least from the legal point of view).

5.3.2 Proposed approach

To face some of the challenges above described, this section describes the implementation of the functional motion planning architecture in control sharing applications. Figure 5.16 presents the proposed stage in addition to INRIA base architecture for automated vehicles (where contributions

[3] Autonomated – Self-Driving vehicles legislation, online at:
http://www.ncsl.org/research/transportation/autonomous-vehicles-legislation.aspx#NCSL Publications

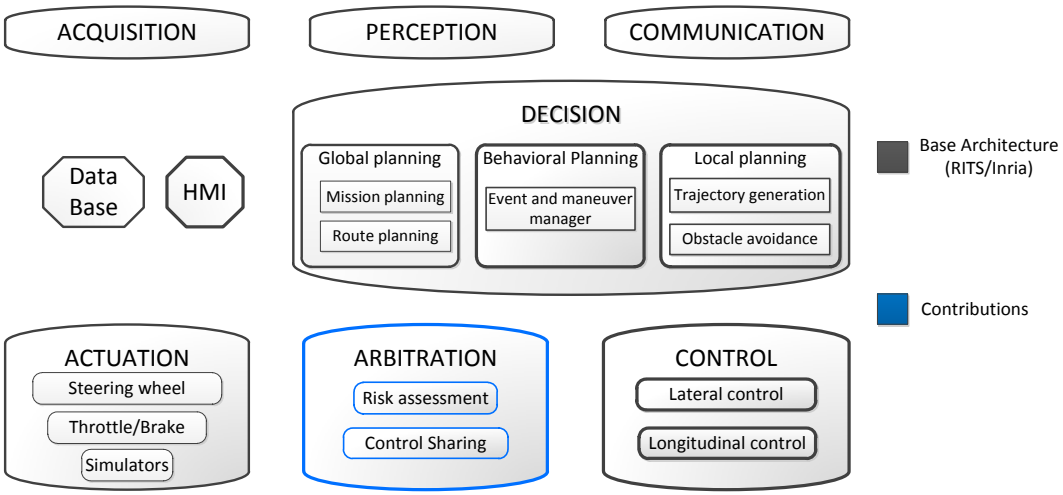


Figure 5.16: Integration of the arbitration module for shared control applications in the base architecture (including contributions from previous chapters)

from previous chapters in the decision stage are already considered). The arbitration stage contains two modules for shared control applications, which allow to have an evaluation of the situation in hand and a decision over the authority each decision maker has at a given moment.

The system relies on the proposed functional planning architecture, which provides the suitable trajectory in real-time, as well as it defines a driving corridor for the driver to follow. Figure 5.17 presents the framework proposed for the arbitration stage. Two main modules are proposed: 1) The risk assessment module determines the risk of the situation in hand with respect to TTC and TTLC; and 2) The sharing controller provides the level of authority with respect to the computed risk.

Once TTC and TTLC are computed, the risk assessment module calculates the collision risk of the situation in hand. However, since the risk is a subjective value, the module implements fuzzy logic, which enables the insertion of human knowledge through its membership functions and is able to provide an answer in real-time . Then, the risk is passed over to the sharing controller module, where the inputs of both decision makers are considered. The common language between the system and the driver will be through haptic signals in the steering wheel. The control input from both, the driver and the automated system, is the angle desired on steering wheel and the level of authority is the haptic force applied (i.e. the higher the authority level, the stronger the signal sent to the steering wheel). The sharing controller module output these haptic signals, establishing a communication bridge between decision-makers at all times. The main contribution of this section are:

- Risk assessment of the situation in hand, considering the TTC and the TTLC, implementing a fuzzy inference system.
- Establish a communication bridge between the driver and the automated system, where both actuate over the steering wheel, sharing the control with smooth authority changes (smooth changes in haptic forces).
- Validate the generality and modularity of the functional planning architecture in shared control applications.

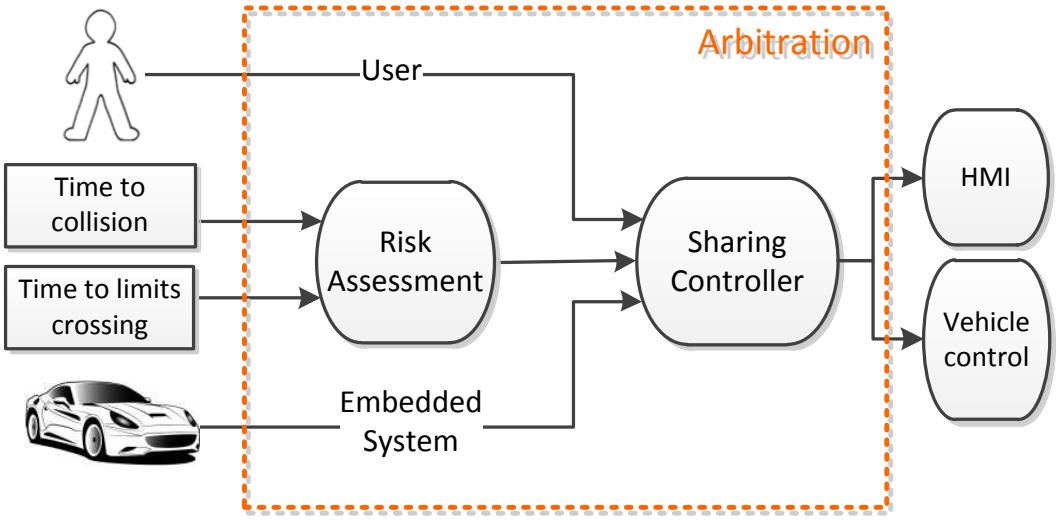


Figure 5.17: Arbitration framework for shared control applications

5.3.3 Risk assessment

The first step to share the control between the driver and the automated system is to assess the risk of the situation in hand. This will allow the system to assess the level of danger of the current configuration of the ego-vehicle and arbitrate whether there is a need for an action coming from the embedded system, and if so, the strength of such intervention. The proposed approach implements two measurements to assess the level of risk of the current state of the vehicle: The TTC and the TTLC as follows:

- **TTC:** The time to collision with other obstacles is described in Chapter 4, where its mathematical representation is as presented in Equation 4.2. In shared control applications, this TTC measure permits to assess the risk of collision with respect to the closest obstacle in conflict with the trajectory of the ego-vehicle.

- **TTLC:** The time to limits crossing concept is similar to Time to Line Crossing (TLC) implemented for LKAS systems [Ishida and Gayko, 2004], [Mammar et al., 2006]. However, in contrast to TLC, the TTLC value computes the time in which the ego-vehicle will exit the driving area determined by the planning stage—which may differ from the lanes in the road when, for example, avoiding obstacles. It permits to assess the risk of going out of bounds, where the boundaries are defined with respect to the predefined trajectory (i.e. the driving area considers lanes but is not constrained by them).

These two variables allow the system to assess the risk associated to the current situation by predicting future collisions and out of bounds configurations. The TTC assess the risk in the longitudinal axis of the ego-vehicle, while the TTLC assess lateral risks of trespassing the boundaries. These two variables cover both actions that the driver can perform in a vehicle (acceleration/deceleration with the pedals and direction changes with the steering wheel). With the implementation of these temporal information, the system can smoothly compute the development of an increasing risky situation and, thus, increase the authority of the system in a smooth way.

5.3.3a Driving boundaries

In control sharing applications, is important to assists the driver when needed and not all the time. The guidance hypothesis [Schmidt and Wulf, 1997] points out that a continuous feedback is not only prejudicial for the learning process, but also subjects that knew the task and went under the effects of continuous guidance tend to perform worse when the guidance is removed. This is because the modified input-output relationship of the problem in hand makes the subjects (humans) reliant on feedback and when this is not present, the performance worsens. Moreover, [Abbink et al., 2012] states that although humans will normally accept that the embedded system is interacting with them and partially taking over the driving task, in some cases the human wants to feel free and in complete control.

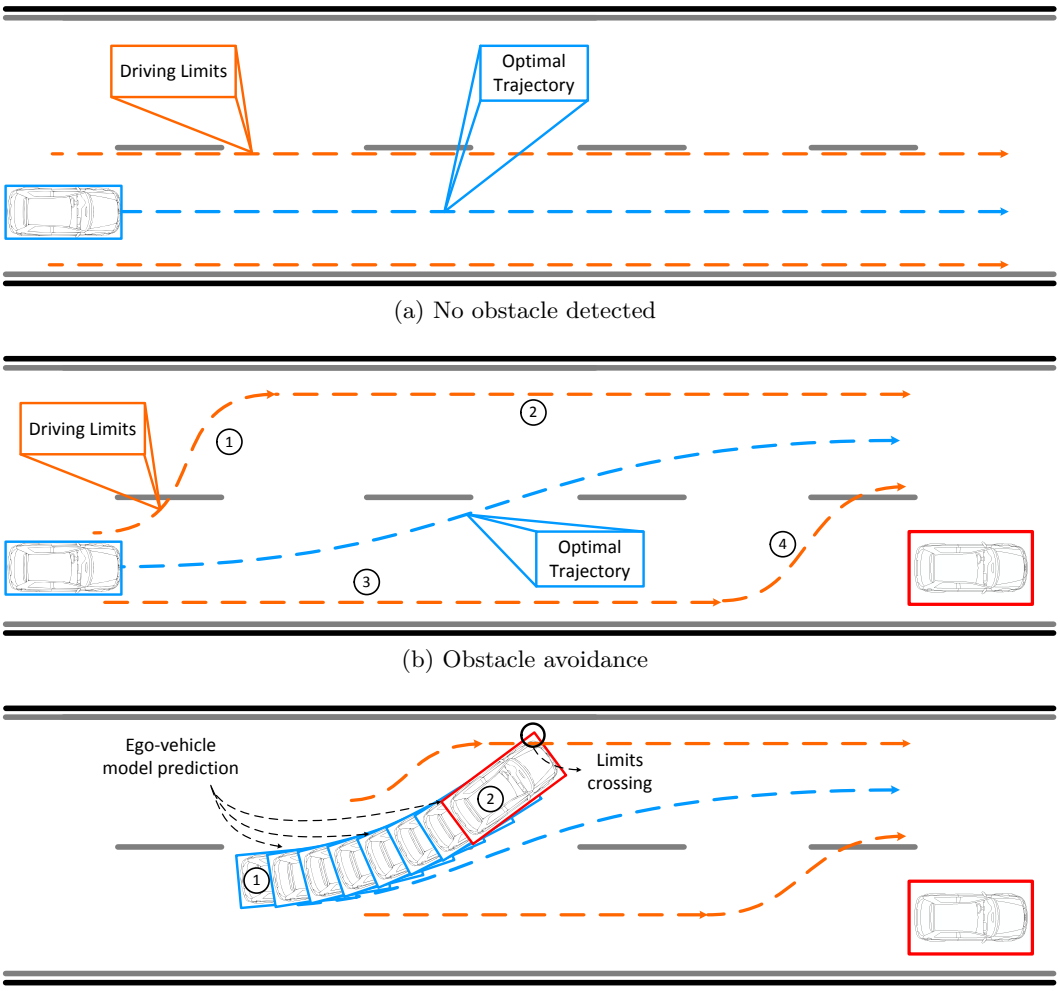
As a way to provide drivers with a sense of driving liberty within the action of shared control and avoid tampering the input-output relationship in the driving task, it is important to define the boundaries for the control to be shared. The aim is to define the correct situations where increasing the authority of automation (and thus the guidance feedback) is helpful, while keeping the human driver skills intact. To do this, the proposed approach defines the driving boundaries, where the TTLC represents the time to go out of bounds. The idea is to provide an obstacle-free corridor, where the driver can decide its own trajectory and, in case risk increases, send haptic signals to inform the driver of the decision of the automated system.

Figure 5.18 depicts the generation of the driving limits, with and without considering obstacles. When no obstacle is in conflict with the initial trajectory, the boundaries are set to the width of the lane as in Figure 5.18a. Here, the optimal trajectory (the path proposed by the automated system) is depicted in blue, whereas the limits of the road are shown in orange. The information for the road width comes from the singular points which define the road layout.

When obstacles are detected and the decision of an avoidance maneuver is taken, the driving boundaries are generated as in Figure 5.18b. In these situations a lane change is needed and the lane boundaries are not considered anymore (defining a driving area which permits the obstacle avoidance maneuver). The generation of the left boundary requires to identify the earliest trajectory the vehicle can take to safely perform the lane change. The generation of this limit trajectory is depicted at the first marker of Figure 5.18b. The trajectory is still feasible but less comfortable and determines the maximum lateral action for an early lane change with respect to the current ego-vehicle configuration (position, speed, among others). It is limited by the road's left boundary. The rest of the left boundary is shown in marker two of Figure 5.18b, where the boundary is coherent with the left physical bound of the road.

The third marker in Figure 5.18b describes the straight segment of the right bound of the driving area. It mimics the right physical bound of the road. Finally, the fourth marker describes the last action for an overtaking maneuver to succeed and describes a late action taken by a driver to avoid the slower obstacle. This last boundary completes the drivable area with respect to the optimal trajectory generated by the embedded system. The boundaries depicted by markers one and four, are created as 5th degree Bézier curves, following the same process explained in Section 4.2.1b.

Once the driving area is set, it is possible to compute future ego-vehicle states and estimate the TTLC. However, the assumption that the ego-vehicle will follow the optimal path proposed by the automation no longer holds. This is because the proposed approach for control sharing assumes that the driver is the one in charge of the driving task and he/she might not necessarily follow this path. Then, the future configurations of the ego-vehicle are computed with the vehicle model described in Equation 5.1. With it, the prediction of future configuration states is done by considering constant velocity and steering angle, as presented in Figure 5.18c. The ego-vehicle



(a) No obstacle detected

(b) Obstacle avoidance

(c) Ego-vehicle predicted trajectory

Figure 5.18: Driving limits generation and evaluation for TTLC

model prediction results from the plot of the trajectory described by Equation 5.1 in 50 steps of $0.1s$, i.e. the future trajectory predicted up to $5s$.

To compute the TTLC, a comparison between predicted configurations and the boundaries is done to detect if the ego-vehicle is going out of bounds and where does it intersect the boundaries. The problem can be posed as the point-into-polygon problem explained in Section 4.2.1a. However, the goal is to find a configuration where the vehicle intersects the polygon formed by the driving boundaries. If this happens, the ego-vehicle predicted path is to go out of bounds within the $5s$ time. The proposed approach evaluates the four points defining the ego-vehicle to see if, at every predicted configuration, the points are within the polygon. The search for a future intersection with the boundaries is stopped as soon as it is detected, where the TTLC is the time the ego-vehicle would take to get to the intersection configuration (i.e. the time for the ego-vehicle to go from position one to position two in Figure 5.18c). If no intersection configuration is detected, then the TTLC will be the maximum value (i.e. $5s$). The process to compute the TTLC is done in real-time, updating the velocity and the steering angle information every time-step.

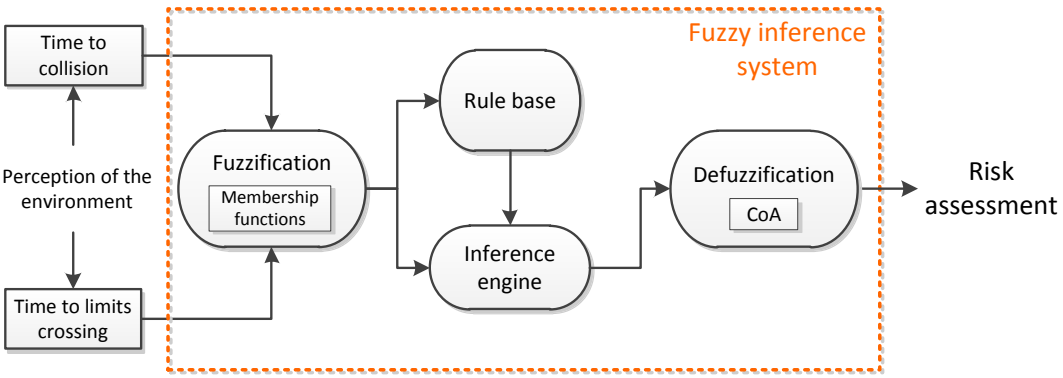


Figure 5.19: Risk assessment flowchart, implementing a fuzzy inference system.

5.3.3b Fuzzy inference system

The risk assessment module takes the TTLC and the TTC to provide a metric of the risk in the current situation. This process is done through a fuzzy inference system. The main reason for using this approach is the need for a model base on human expert knowledge of the current situation to effectively assess the risk. The complexity of the driving task and the unpredictable behavior of drivers makes difficult the modeling of the current situation and the risk associated to it. Fuzzy methods provide an alternative way to deal with increasing complexity by inserting expert procedural knowledge in the inference process, aiding in the estimation of risk.

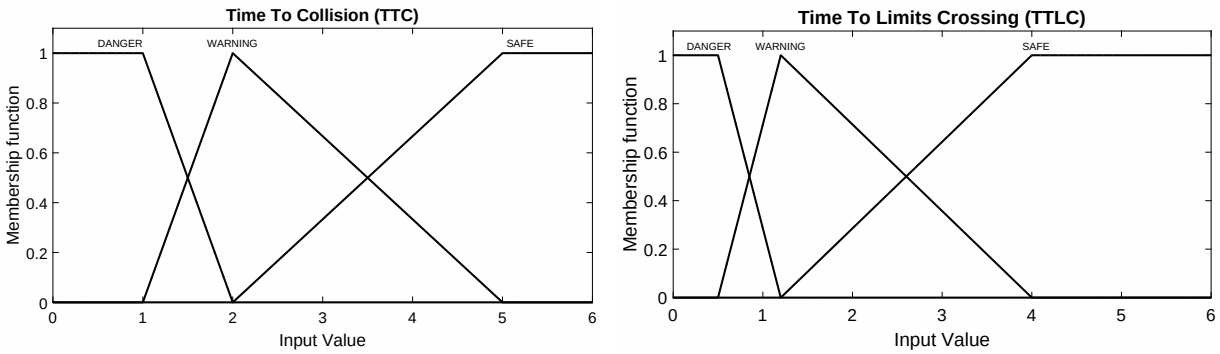
A fuzzy system is composed of three main stages [Naranjo et al., 2008], [Mamdani, 1974]: 1) Fuzzification: in this step, the crisp inputs entering the fuzzy system are translated to linguistic values. This depends on the membership functions which represent the human driver's subjective knowledge. In the proposed approach, the inputs are the TTC and the TTLC (see Figure 5.19); 2)Inference engine: the linguistic values are evaluated with respect to the fuzzy rule base presented in Table 5.1. This nine rules contains the expert knowledge to evaluate the situation and provide a level of risk, given the values of inputs; and 3)Defuzzification: this stage implements the Center of Area (CoA) method to provide an average output from the conclusion of each rule given in the inference process, i.e. the output information goes from fuzzy values to crisp values, giving a metric for the risk of the situation.

The output of the system has five linguistic labels (i.e. Low, MidLow, Medium, MidHigh and High), modeled as singleton values within the [0,1] interval. The risk level is also given in the [0,1], representing complete safety when its value is zero and highest danger when its value is one. The CoA mathematical representation is as described in Equation 5.3,

$$Risk = \frac{\sum w_i O_i}{\sum w_i} \tag{5.3}$$

where w_i represents the weight assigned to each rule evaluation in the inference process. The O_i term represents the singletons values assigned to output linguistic values. Then defuzzification stage gives as a result the "Risk" term.

The membership functions implemented for the TTC and TTLC input values are shown in Figure 5.20. They represent the human subjective knowledge with respect to the risk associated to each input. For TTC (see Figure 5.20a) three linguistic labels have been assigned (Safe, Warning and Danger). The safe membership region covers the interval [2.0s,∞), where the TTC is considered safe. The warning region is set within the interval of [1s,5s] indicating the beginning of



(a) (b)

Figure 5.20: Membership function definition.

the unsafe region. The danger membership covers from [0s,1s] indicating the critical region for TTC values and suggesting a high risk when evaluated in rules presented in Table 5.1. The TTC value is constrained to positive values. In case a negative value appears, meaning the obstacle is getting away from the vehicle, it is taken as safe since no collision is expected. The labels defined for the TTLC are similar, however the membership functions differ slightly: the safe region is defined in the interval [1.2s,∞), the warning area is defined as the interval [0.5s,4s] and the danger membership function is defined at the [0s,1.2s] interval.

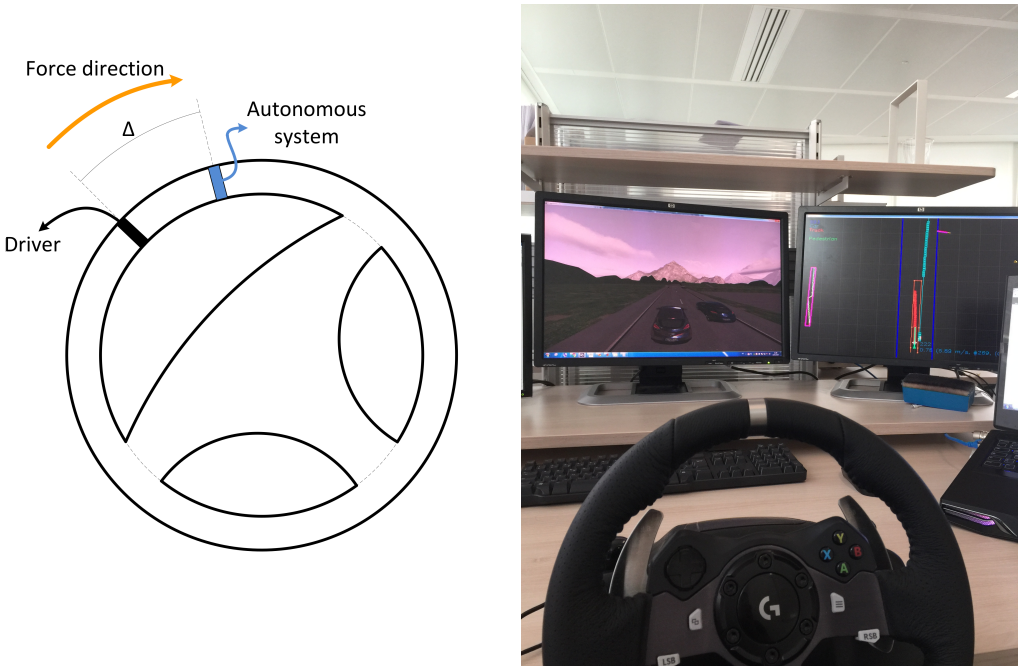
Table 5.1: Fuzzy inference rule base

Rule	Linguistic formulation
R1	IF *TTC* **Safe** AND *TTLC* **Safe** THEN *Risk* **Low**
R2	IF *TTC* **Warning** AND *TTLC* **Safe** THEN *Risk* **MidLow**
R3	IF *TTC* **Danger** AND *TTLC* **Safe** THEN *Risk* **MidLow**
R4	IF *TTC* **Safe** AND *TTLC* **Warning** THEN *Risk* **MidLow**
R5	IF *TTC* **Warning** AND *TTLC* **Warning** THEN *Risk* **Mid**
R6	IF *TTC* **Danger** AND *TTLC* **Warning** THEN *Risk* **MidHigh**
R7	IF *TTC* **Safe** AND *TTLC* **Danger** THEN *Risk* **MidHigh**
R8	IF *TTC* **Warning** AND *TTLC* **Danger** THEN *Risk* **High**
R9	IF *TTC* **Danger** AND *TTLC* **Danger** THEN *Risk* **High**

5.3.4 Shared control

Once the risk of the situation has been computed by the fuzzy inference system, the control can be shared with the user through haptic signals over the steering wheel. The aim is to create a communication bridge between both decision makers, closing the loop by including the driver in the automated task. Although the system is initially in supervision mode, it will intervene when risk is high, by sending the haptic signals to the steering wheel.

For this, a simulation environment has been design including a Logitech G920 steering wheel that communicates with the automated system and allows to insert the driver in the loop, as shown in Figure 5.21b. This simulation setup allow the control of the steering wheel by the automated



(a) Force feedback example (b) Haptic HMI and simulation setup

Figure 5.21: Haptic HMI and force feedback for shared control applications

system through force feedback. This force acts as guidance cues for the driver to know the best steering input computed by the system.

The behavior of the steering wheel is modeled as an string or coil, based mathematically over Hooke's law. The system provides a force feedback only when there is a disagreement between the embedded system and the driver. Figure 5.21a presents a graphic view of the system. Here, the input received from the driver is in black, as the direction of the steering wheel. The decision for the steering angle coming from the embedded system is presented in blue, and the distance between each decision is given as an angular distance "Δ". Equation 5.4 describes the operation range for the feedback to be given, as well as the mathematical description of the system, considering that the steering wheel is set to turn only within the interval $[-100°, 100°]$.

$$F_{(k,\Delta)} = \begin{cases} k, & \text{if } \Delta > 5° \\ \frac{k\Delta}{5°}, & \text{if } -5° \leq \Delta \leq 5° \\ -k, & \text{if } \Delta < -5° \end{cases} \tag{5.4}$$

where $F_{(k,\Delta)}$ is the force feedback dependent on the stiffness coefficient and the angular distance between the decision makers. Here, Δ varies within a range of 5° and belongs to the interval $[-5°, 5°]$. The stiffness coefficient (characteristic to the material of which the spring is made in Hooke's law), is governed by the risk and takes its value within the $[0, 1]$ interval. The force feedback is passed to the actuator (the steering wheel, see Figure 5.21b) as a percentage, i.e. $F_{(k,\Delta)}$ takes its value also within the [0,1] interval. Then, the actuator is in charge of turning this force index to torque in the steering wheel and, thus, establishing a communication bridge with the driver.

5.3.5 Validation tests

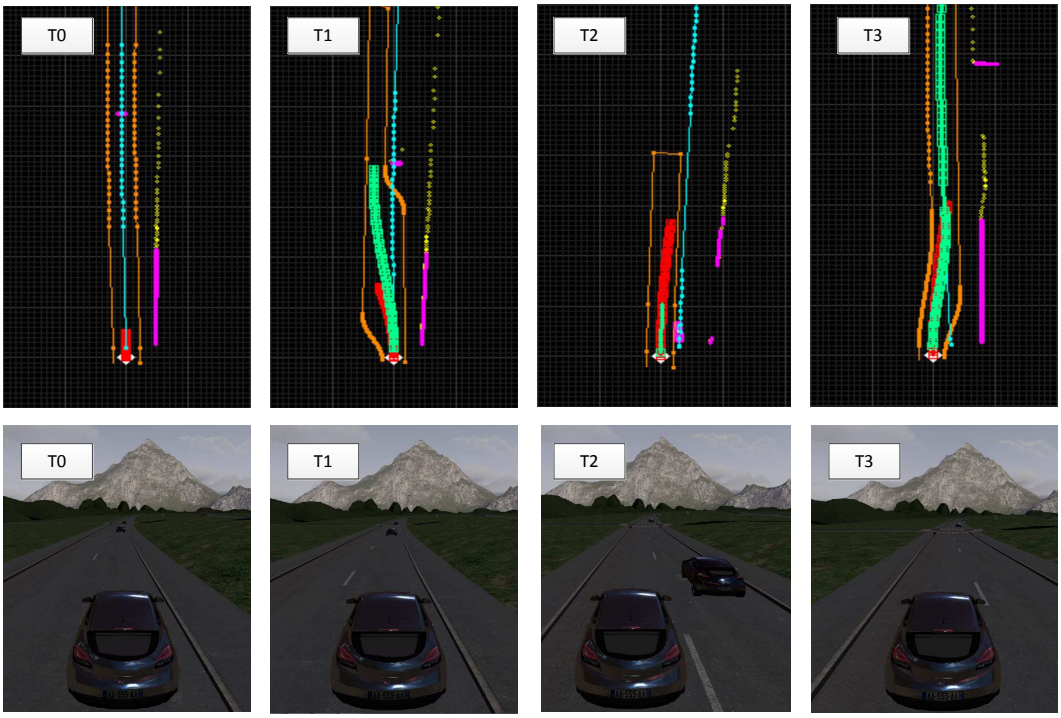
The setup presented in Figure 5.21b implements Pro-SiVIC as the simulation environment to emulate the vehicle behavior and RTMaps as a thread management software, where the functional architecture is implemented along with the haptic HMI. This way, the driver can be included in the loop of control. The validation test is performed in a simulated urban scenario with tree vehicles (one as the ego-vehicle and the other two representing static and dynamic obstacles). The ego-vehicle evaluates the TTC and the TTLC to create a risk assessment measurement that can determine the level of control and responsibility of each decision-maker. The scenario designed for the test is similar to the one simulated in Figure 5.7a, only this time the driver is considered within the control loop. In the path, two obstacles (one static and one dynamic) are shown and the driver is to drive in the right lane, taking the left one only to avoid obstacles.

The scenario of the first avoidance maneuver is as shown in Figure 5.22a. The ego vehicle is represented by the blue vehicle in the bottom pictures and by white cross at the bottom of each top figure. The leftmost figure (T0) depicts the generation of the driving limits and the optimal path when no obstacles are detected. The lateral limits of the road are shown in orange, the original path in teal, the driver's predicted trajectory in red (up to five seconds) and obstacles from laser impacts in magenta. The middle left figure (T1) presents the initiation of an avoidance maneuver because an obstacle has been detected to have a future collision state with the ego-vehicle. The avoidance path is in green and the limits (in orange) have changed to a configuration which lets the driver a free action region to drive. This configuration permits the driver to have some freedom with respect to the path to follow when overtaking, but keeping him/her in safety, and sending haptic signals when the TTLC reaches a dangerous levels (the analysis of the red predicted trajectory and the orange limits allow to compute the TTLC).

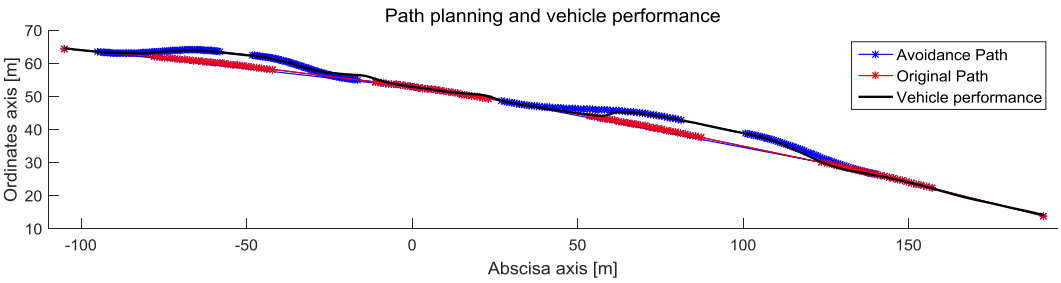
Middle right Figure (T2) presents the passing stage and the limits for the passing maneuver, where the obstacle is still in front of the ego-vehicle (on the right lane), represented with a magenta line in conflict with the original teal path. Rightmost figure (T3) shows the planned maneuver to return to the path in green, where the orange limits slightly differ from the avoidance stage of the overtaking. The limit to the right of the ego-vehicle depicts the fastest route back. However, the limit to the left is a projection of the returning path the driver has to follow. This allows to encourage a fast return to the original lane, while keeping the free driving area (avoiding to be too intrusive) and a high comfort. The limits in the avoidance and returning phase of the overtaking maneuver are set to have a curvature that complies with the lowest level of comfort (i.e. $2.5m/s^2$), they thus determine the limit trajectories to move from one lane to the other while avoiding an obstacle or returning to the original lane.

Figure 5.22b presents the optimal path generated for the vehicle in red, the optimal collision free path in blue and the vehicle behavior in black. Real-time behavior of the whole planning architecture is ensured, even when adding the generation of the limits for the safe driving area. This is shown in Figure 5.22c, where the computation of the avoidance path and the continuous computation of the driving limits is lower than $60ms$. The two overtaking maneuvers are visible between $7s$-$28s$ and $29s$-$48s$ respectively, where the time increases at the beginning and at the end of the maneuvers because external and internal driving limits are being computed; and decreases when the path is only duplicated to the right and the left as in passing stages or when no obstacle is detected (no overtaking maneuver is in process).

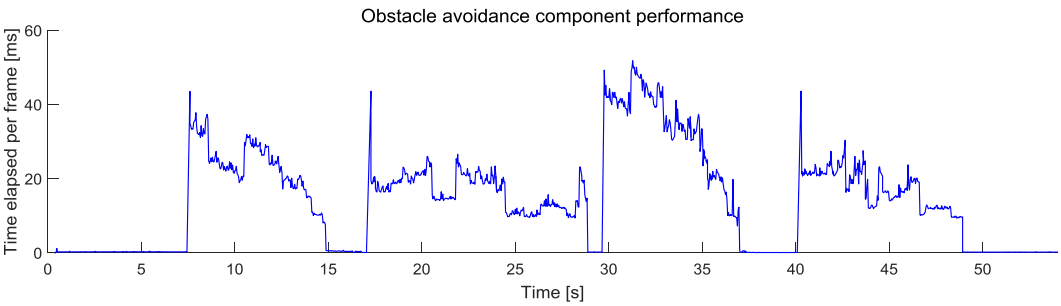
The performance of the driver does not necessary follows the optimal collision free path, but stays always close to its center (around $0.5m$ of lateral deviation and less than $0.2rad$ of angular deviation, see Figure 5.23a). However the driver deviates more than $1m$ in three opportunities: 1) between $20s$ and $25s$; 2) between $30s$ and $35s$; and 3) between $42s$ and $46s$. Figure 5.23b depicts



(a) Scenario representation from a bird view perspective (upper figures), including: driving road limits (orange), original trajectory (teal), avoidance trajectory (green), driver's trajectory prediction (red), obstacles (magenta). Lower figures represent the same temporal interval in a camera view mode.



(b)



(c)

Figure 5.22: Path generation in shared control applications.

low values when the vehicle gets away from the current path and the predicted time to crossing the limits decreases. Figure 5.23c shows the TTC with respect to the obstacle in conflict with the ego-vehicle original path. It is a reference value, taken to assess the risk of the situation in hand, with respect to the remaining time for the ego-vehicle to "collide" with the obstacle if its path goes directly to it at the current relative speed.

Figure 5.23d shows the risk output from the fuzzy inference system in black (output of the risk assessment in Figure 5.17) and depicts the percentage of force given to the steering wheel in blue (output of the sharing controller in Figure 5.17). The force level at the beginning of the path is maximum since the driver is not holding the steering wheel. From second six, the driver takes the steering wheel and the force level decreases, with three seconds to go from the maximum level of torque, to completely passing the control to the driver if the situation is safe (TTC and TTLC are low).

When TTC and TTLC values decrease, the risk and the haptic cues increase, as it is the case in seconds 24 and 29. Second 18 depicts a case where both, the TTC and the TTLC increase the level of risk—and thus the force action over the steering wheel—because both values represent a risky situation of a driver that approaches an obstacle and has a low predicted time to exit the safe trajectory envelopes. At $34s$ the driver is in a critical situation, since he/she approaches an obstacle but remains in the right lane and suddenly releases the steering wheel. The system detects the absence of the driver input and sets the force to the maximum value to gain back the control of the ego-vehicle and redirect it to the optimal path. The risk at this moment peaks at maximum value since the actions taken by the controller produce a small value of TTLC while also having a close obstacle (i.e. TTC is also low). However, the vehicle never goes out of bounds and recovers a normal driving behavior as evidenced in Figure 5.23a. When the driver takes back the control at $41s$, the system smoothly shifts the control back from full automation to manual driving, since no risky situation is found. Finally the driver deviates slightly from the path, creating a drop in the TTLC and a force feedback of half the full potential of the steering wheel guides the driver back to the optimal path before the final stop (at $54s$).

5.4 Conclusion

The chapter presents the validation tests of the functional architecture in different simulated and real urban scenarios. The development software was RTMaps, where the handling of different threads in real-time allows the implementation of the different modules that integrate the embedded architecture. Simulations were carried out in Pro-SiVIC, where urban environments are emulated to provide a realistic behavior of vehicles and different road actors. To validate the algorithms, the functional architecture was first tested in an scenario comprising two turns and two roundabouts of different radii. Results showed the generation of a G^1 continuous path, in real-time, to address each intersection while also addressing singular global points that model straight stretches.

The quintic Bézier speed planner then adds a speed profile to complete the generation of a trajectory that takes the vehicle from point "A" to point "B", considering comfort limits. It considers the curvature profile of the path, avoiding lateral or longitudinal acceleration overshoots that might cause discomfort. Results show the generation of a G^2 continuous speed profile and a good behavior of the simulated vehicle while following the trajectory, only surpassing comfort limits by a factor of 5% of the limit comfort acceleration (mainly due to temporal delays in the vehicle model).

Obstacle avoidance maneuvers were also shown, at constant speeds, where two obstacles are overtaken. The first obstacle was static. The avoidance module generates G^2 continuous paths,

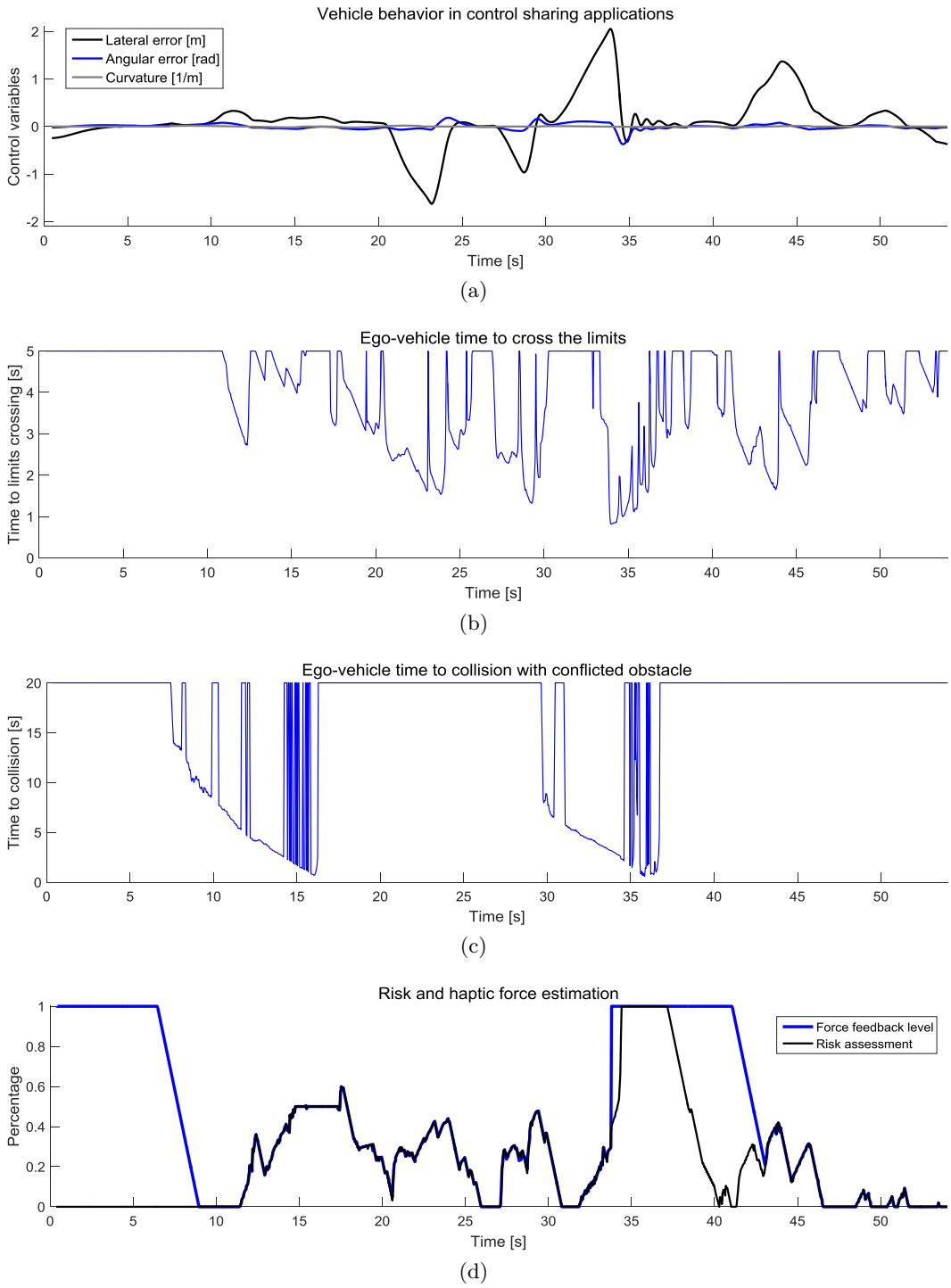


Figure 5.23: Driver performance, risk and force feedback estimation from TTC, TTLC.

implementing the three phase overtaking maneuver. The second obstacle was dynamic and the temporal relation shown in the experiments—between the ego-vehicle and the obstacle—evidences the proper generation of the avoidance maneuver, even when obstacles are moving at constant speeds. The ego-vehicle achieves a good behavior while following the path and complies with

comfort accelerations while returning to the initial optimal path.

Once the approach is validated in simulations, it is then tested in real urban environments at INRIA-Rocquencourt. The platform used to validate the algorithms was the Cyberbus, an electric minibus system without mechanical driver actuators. First, the trajectory planning capabilities were tested with an itinerary comprising two turns, a roundabout and several singular points describing the road layout. The generation of the trajectories is done in real-time and the ego-vehicle is able to navigate up to the destination point with less than $0.25m$ of lateral error, evidencing a good behavior and validating the trajectory generation algorithm. Then, avoidance maneuvers are tested. Static and dynamic obstacles are overtaken by the Cyberbus once it determined a possible future collision, since the obstacles were in the original path to be taken. The new planning strategy of the ego-vehicle is G^2 continuous, generated in real-time and only as needed, optimizing the computation time. Results show a good behavior of the vehicle and validate the approach in real platforms.

The functional architecture is also tested in shared control applications. The computation of the driving limits allows to assess the risk in the situation in hand and effectively arbitrate the level of responsibility between the two decision makers. The system is shown to respond in real-time, where the risk level is determined from the time to collision to the obstacle in front and the time to limits crossing. The risk then transforms into haptic signals on the steering wheel, creating a communication bridge that allows the human driver to understand the indications of the automated system. The haptic feedback is given as needed instead of all the time, where the driver remained free of navigating the road within a safe trajectory envelope. When the driver input is missing from the system, it can go up to fully automated capabilities and take back the vehicle to the safe path, even in avoidance maneuvers. Fuzzy inference systems are suitable for this kind of applications—as validated in the test—enabling the insertion of the human expert knowledge in defining subjective values such as the risk in different driving situations (e.g. going out of bounds, overtaking a slower vehicle, among others).

Chapter 6

Conclusion

Advances in the road transportation field show an important increase in maturity of the technology and, with it, automated vehicles are getting smarter, more robust and closer to reality. Chapter 2 gives a broad overview of these systems, focusing on the motion planning point of view, where is clear that a system capable of generating comfortable trajectories in real-time for automated vehicles motion planning is missing. Moreover, there are still several milestones in the road to fully automated vehicles, such as the social acceptance of the technology, the technological developments and the remaining legal/liability issues.

The present Ph.D. thesis addresses some of the aforementioned challenges by proposing a functional architecture for trajectory planning. It implements interpolation techniques to generate suitable trajectories, including comfort as the main design parameter. It is able to plan, in real-time, G^1 continuous paths for turns and roundabouts, determining the optimal strategy for the vehicle to follow. A novel speed planning algorithm provides optimal G^2 continuous speed profiles that added to the geometrical path, form the appropriated trajectory for the vehicle to execute.

Static and dynamic obstacles are also considered. The proposed functional architecture is capable of detecting future collision states and avoid them by generating a new path in real-time. This new path is generated in three main phases (avoidance, passing and returning to the original lane), only when needed and not all the time, thus optimizing computation time. The path is G^2 continuous, meaning that the overall curvature of this new strategy is continuous in all its domain.

The proposed architecture for automated vehicle's motion planning has been simulated and tested in real platforms. Simulations, in Pro-SiVIC and Matlab, validated in a first time the generation of the optimal trajectories for motion planning in static and dynamic environments. Real tests, performed at Inria's facilities with the Cyberbus, validated the whole architecture in real urban scenarios, showing good results for different itineraries, avoiding obstacles (static and dynamic) when future collisions are detected.

The architecture has also been implemented in shared control applications. It allows to consider the driver in the control loop and arbitrate between the two decision makers (the embedded system and the human driver), according to the situation in hand. The proposed motion planning architecture serves as a basis for this kind of applications since it delivers the optimal path to follow and the driving limits to evaluate the performance of the driver. A fuzzy inference system is able to assess the risk of the situation, by evaluating the TTC and TTLC. It also allows the insertion of expert human knowledge, necessary when estimating subjective values such as the risk in different situations. The functional architecture is able to delimit the area where the driver can safely drive by determining the limits of the road when no obstacles are detected, and to determine the earliest and latest paths for the vehicle to avoid obstacles, once these are detected.

6.1 Contributions to the state of the art

With comfort as the main design parameter, the contributions for each presented chapter are as follows:

6.1.1 Static environments

- Singular points describe the road layout in the global planner, providing the necessary information for the path to follow, the destination, road and intersections configurations, among others.

- Real-time G^1 continuous path generation for urban scenarios using parametric curves. The path generation takes the singular points defined by the global planner and creates local paths to obtain a smooth strategy to navigate the road. Minimal curvature paths, reducing lateral accelerations and increasing comfort.

- G^1 continuous path for roundabout entrance and exit, implementing an improved generation of the parametric control points. The path continuity, curvature minimization and comfort constraints considerations for in-roundabout driving are the most important design criterion. Two-lane roundabouts are considered, including lane-change and taking different exits.

- A novel real-time speed profile generation, based on quintic Bézier curves has been proposed. It complies with comfort constraints in [ISO, 1997] by considering longitudinal, lateral and total acceleration limits. This allows the generation of the optimal trajectory (the geometrical path and the speed profile) for the vehicle to follow.

6.1.2 Dynamic environments

- Generation of a polygon representing future ego-vehicle's positions, allowing real-time detection of future collision states from perception and vehicle state information.

- The curve generation is done complying with comfort constraints, generating a G^2 continuous avoidance path by implementing quintic Bézier curves. This greatly improves the controllers response while following the path and reduces lateral accelerations, thus, increasing comfort.

- The path generation process is done in real-time, always under a frame rate of $60ms$, allowing the fast generation of a new planning strategy to avoid static and dynamic obstacles in urban environments.

- The planning of these new curvature continuous path is done as needed instead of all the time. This is possible since an understanding of the situation is done a priori, saving processing time.

6.1.3 Shared Control

- The generation of the driving limits (based on the planning architecture proposed in previous chapters) gives the possibility to create an obstacle free corridor, where the human can drive without interruption from the system, thus avoiding too much intrusion, specially in overtaking maneuvers.

- The proposed method to compute the risk assessment implements a fuzzy inference system, capable of introducing human expert knowledge for the computation of subjective values such as risk. It is possible by analyzing the TTC and the TTLC.
- Establish a communication bridge between the driver and the automated system, where both actuate over the steering wheel, sharing the control with smooth authority changes (smooth changes in haptic force levels).
- The application of the functional architecture within a control sharing ADAS validates the generality and modularity of the functional planning architecture and raises new opportunities for the transition between automated driving and autonomous vehicles.

6.2 Future research direction

- **Motion planning:** So far, the motion planning problem has been addressed from the path evaluation point of view, instead of understanding the scene first and planning later. This is visible with the important amount of methods that search exhaustively a path in a network of nodes (A* family), randomly (RRT family), evaluating several curves from a database (Spatio-temporal Lattices and some interpolation approaches), among others. If it is true that the heavy computation aspect of planning cannot be avoided, the present Ph.D. thesis proposes a framework about how the scene could be rapidly analyzed and the new planning strategy can rely on the understanding of the scene, instead of evaluating a predetermined set of curves and discarding those in conflict with obstacles, the road layout, etc. In other words, the obstacle free path could be determined before optimizing the trajectory to be taken.

  The development of intelligent systems, capable of understanding the scene after the perception stage "tracks" all obstacles and determines the road layout, is necessary. Today, these systems are still missing from the base architecture of automated vehicles. Since the driving task is above all a social interactive task, it is necessary to develop such systems to properly interact in complex environments such as urban scenarios.

- **Shared control and arbitration:** Control sharing applications are the clear next step between nowadays ADAS and highly automated vehicles. They permit a smooth transition from one to another while tackling the social barrier, keeping the driver in the loop of control and avoiding the legal bottleneck since the driver is always in control of the vehicle. Technological development profits also from this, since the systems can be thoroughly tested and have the human response as ground truth, specially learning systems that need labeled data to increase its knowledge base. This Ph.D. work has presented a first approach about how to deal with driver-in-the-loop automated control sharing systems. Future research directions will focus on extracting more driver patterns to determine whether the automated system should take the control or not with higher accuracy, leading to complex artificial intelligence-based algorithms for managing all the given information, increasing algorithm robustness and effectiveness.

Bibliography

[Abaza and Hussein, 2009] Abaza, O. A. and Hussein, Z. S., 2009. Comparative analysis of multi-lane roundabout capacity "case study". In: IEEE 70th Vehicular Technology Conference Fall (VTC 2009-Fall), pp. 1–5.

[Abbink, 2006] Abbink, D. A., 2006. Neuromuscular analysis of haptic gas pedal feedback during car following. TU Delft, Delft University of Technology.

[Abbink et al., 2012] Abbink, D. A., Mulder, M. and Boer, E. R., 2012. Haptic shared control: smoothly shifting control authority? Cognition, Technology & Work 14(1), pp. 19–28.

[Aeberhard et al., 2015] Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F., Huber, W. and Kaempchen, N., 2015. Experience, results and lessons learned from automated driving on germany's highways. Intelligent Transportation Systems Magazine, IEEE 7(1), pp. 42–57.

[Anderson et al., 2014] Anderson, J. M., Nidhi, K., Stanley, K. D., Sorensen, P., Samaras, C. and Oluwatola, O. A., 2014. Autonomous Vehicle Technology: A Guide for Policymakers. Rand Corporation.

[Anderson et al., 2012] Anderson, S. J., Karumanchi, S. B. and Iagnemma, K., 2012. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. In: Intelligent Vehicles Symposium (IV), 2012 IEEE, IEEE, pp. 383–388.

[Aoude et al., 2010] Aoude, G. S., Luders, B. D., Levine, D. S. and How, J. P., 2010. Threat-aware path planning in uncertain urban environments. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp. 6058–6063.

[Bacha et al., 2008] Bacha, A., Bauman, C., Faruque, R., Fleming, M., Terwelp, C., Reinholtz, C., Hong, D., Wicks, A., Alberi, T., Anderson, D. et al., 2008. Odin: Team victortango's entry in the darpa urban challenge. Journal of Field Robotics 25(8), pp. 467–492.

[Bai et al., 2009] Bai, Y., Xue, K. and Yang, X., 2009. Block mechanism of left-turned flow at signal-controlled roundabout. In: WRI Global Congress on Intelligent Systems, Vol. 3, pp. 443–449.

[Baltzer et al., 2014] Baltzer, M., Flemisch, F., Altendorf, E. and Meier, S., 2014. Mediating the interaction between human and automation during the arbitration processes in cooperative guidance and control of highly automated vehicles. In: Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE.

[Behringer and Muller, 1998] Behringer, R. and Muller, N., 1998. Autonomous road vehicle guidance from autobahnen to narrow curves. Robotics and Automation, IEEE Transactions on 14(5), pp. 810–815.

[Beiker, 2012] Beiker, S. A., 2012. Legal aspects of autonomous driving. Santa Clara L. Rev. 52, pp. 1145.

[Berglund et al., 2010] Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M. and Soderkvist, I., 2010. Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. Automation Science and Engineering, IEEE Transactions on 7(1), pp. 167–172.

[Bertozzi et al., 2011] Bertozzi, M., Bombini, L., Broggi, A., Buzzoni, M., Cardarelli, E., Cattani, S., Cerri, P., Coati, A., Debattisti, S., Falzoni, A. et al., 2011. Viac: An out of ordinary experiment. In: Intelligent Vehicles Symposium (IV), 2011 IEEE, IEEE, pp. 175–180.

[Bohren et al., 2008] Bohren, J., Foote, T., Keller, J., Kushleyev, A., Lee, D., Stewart, A., Vernaza, P., Derenick, J., Spletzer, J. and Satterfield, B., 2008. Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge. Journal of Field Robotics 25(9), pp. 598–614.

[Bosetti et al., 2014] Bosetti, P., Da Lio, M. and Saroldi, A., 2014. On the human control of vehicles: an experimental study of acceleration. European Transport Research Review 6(2), pp. 157–170.

[Bosetti et al., 2015] Bosetti, P., Da Lio, M. and Saroldi, A., 2015. On curve negotiation: From driver support to automation. IEEE Transactions on Intelligent Transportation Systems 16(4), pp. 2082–2093.

[Bouraoui et al., 2011] Bouraoui, L., Boussard, C., Charlot, F., Holguin, C., Nashashibi, F., Parent, M. and Resende, P., 2011. An on-demand personal automated transport system: The citymobil demonstration in la rochelle. IEEE Intelligent Vehicles Symposium 4, pp. 1086 – 1091.

[Boverie et al., 2011] Boverie, S., Cour, M. and Le Gall, J. Y., 2011. Adapted human machine interaction concept for driver assistance systems driveasy. In: World Congress, Vol. 18, pp. 2242–2247.

[Braid et al., 2006] Braid, D., Broggi, A. and Schmiedel, G., 2006. The terramax autonomous vehicle. Journal of Field Robotics 23(9), pp. 693–708.

[Brezak and Petrovic, 2014] Brezak, M. and Petrovic, I., 2014. Real-time approximation of clothoids with bounded error for path planning applications. Robotics, IEEE Transactions on 30(2), pp. 507–515.

[Broggi et al., 1999] Broggi, A., Bertozzi, M., Fascioli, A., Bianco, C. G. L. and Piazzi, A., 1999. The argo autonomous vehicle's vision and control systems. International Journal of Intelligent Control and Systems 3(4), pp. 409–441.

[Broggi et al., 2013a] Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M., Medici, P. and Versari, P., 2013a. Extensive tests of autonomous driving technologies. Intelligent Transportation Systems, IEEE Transactions on 14(3), pp. 1403–1415.

[Broggi et al., 2014] Broggi, A., Cerri, P., Debattisti, S., Laghi, M. C., Medici, P., Panciroli, M. and Prioletti, A., 2014. Proud-public road urban driverless test: Architecture and results. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, pp. 648–654.

[Broggi et al., 2013b] Broggi, A., Debattisti, S., Panciroli, M. and Porta, P. P., 2013b. Moving from analog to digital driving. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 1113–1118.

[Broggi et al., 2010] Broggi, A., Medici, P., Cardarelli, E., Cerri, P., Giacomazzo, A. and Finardi, N., 2010. Development of the control system for the vislab intercontinental autonomous challenge. In: Proc. 13th Int Intelligent Transportation Systems (ITSC) IEEE Conf, pp. 635–640.

[Broggi et al., 2012] Broggi, A., Medici, P., Zani, P., Coati, A. and Panciroli, M., 2012. Autonomous vehicles control in the vislab intercontinental autonomous challenge. Annual Reviews in Control 36(1), pp. 161 – 171.

[Chen et al., 2008] Chen, Y.-L., Sundareswaran, V., Anderson, C., Broggi, A., Grisleri, P., Porta, P. P., Zani, P. and Beck, J., 2008. Terramax™: Team oshkosh urban robot. Journal of Field Robotics 25(10), pp. 841–860.

[Choi et al., 2008] Choi, J.-w., Curry, R. and Elkaim, G., 2008. Path planning based on bézier curve for autonomous ground vehicles. In: World Congress on Engineering and Computer Science 2008, WCECS'08. Advances in Electrical and Electronics Engineering-IAENG Special Edition of the, IEEE, pp. 158–166.

[Chu et al., 2012] Chu, K., Lee, M. and Sunwoo, M., 2012. Local path planning for off-road autonomous driving with avoidance of static obstacles. Intelligent Transportation Systems, IEEE Transactions on 13(4), pp. 1599–1616.

[Coombs et al., 2000] Coombs, D., Murphy, K., Lacaze, A. and Legowik, S., 2000. Driving autonomously off-road up to 35 km/h. In: Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE, pp. 186–191.

[Cremean et al., 2006] Cremean, L. B., Foote, T. B., Gillula, J. H., Hines, G. H., Kogan, D., Kriechbaum, K. L., Lamb, J. C., Leibs, J., Lindzey, L., Rasmussen, C. E. et al., 2006. Alice: An information-rich autonomous vehicle for high-speed desert navigation. Journal of Field Robotics 23(9), pp. 777–810.

[Da Lio et al., 2015] Da Lio, M., Biral, F., Bertolazzi, E., Galvani, M., Bosetti, P., Windridge, D., Saroldi, A. and Tango, F., 2015. Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems. IEEE Transactions on intelligent transportation systems 16(1), pp. 244–263.

[De Brabander and Vereeck, 2007] De Brabander, B. and Vereeck, L., 2007. Safety effects of roundabouts in flanders: Signal type, speed limits and vulnerable road users. Accident Analysis and Prevention.

[de Winter et al., 2008] de Winter, J. C., Mulder, M., Van Paassen, M., Abbink, D. A. and Wieringa, P. A., 2008. A two-dimensional weighting function for a driver assistance system. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38(1), pp. 189–195.

[Dokic et al., 2015] Dokic, J., Müller, B. and Meyer, G., 2015. European roadmap smart systems for automated driving. European Technology Platform on Smart Systems Integration.

[Dolgov and Thrun, 2009] Dolgov, D. and Thrun, S., 2009. Autonomous driving in semi-structured environments: Mapping and planning. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 3407–3414.

[Dolgov et al., 2010] Dolgov, D., Thrun, S., Montemerlo, M. and Diebel, J., 2010. Path planning for autonomous vehicles in unknown semi-structured environments. The International Journal of Robotics Research 29(5), pp. 485–501.

[Dopart, 2015] Dopart, K., 2015. U.s. dot automation program, at: http://www.its.dot.gov/presentations/pdf/trb2015_vha_committee_dopart.pdf.

[Durekovic and Smith, 2011] Durekovic, S. and Smith, N., 2011. Architectures of map-supported adas. In: Intelligent Vehicles Symposium, pp. 207–211.

[Elbanhawi and Simic, 2014] Elbanhawi, M. and Simic, M., 2014. Sampling-based robot motion planning: A review. Access, IEEE 2, pp. 56–77.

[Erlien et al., 2016] Erlien, S. M., Fujita, S. and Gerdes, J. C., 2016. Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. IEEE Transactions on Intelligent Transportation Systems 17(2), pp. 441–451.

[ERTRAC, 2015] ERTRAC, E. R. T. R. A. C., 2015. Automated driving roadmap.

[European Commision, 2015] European Commision, E. U., 2015. Road safety in the european union: Trends, statistics and main challenges. Technical report, Directorate-General Mobility and Transport.

[European Commision, 2016] European Commision, E. U., 2016. Declaration of amsterdam, co-operation in the field of connected and automated driving.

[Farouki, 2008] Farouki, R. T., 2008. Pythagorean—Hodograph Curves. Springer.

[Federal Automated Vehicles Policy, 2016] Federal Automated Vehicles Policy, 2016. U.S. Department of Transportation.

[Ferguson and Stentz, 2006] Ferguson, D. and Stentz, A., 2006. Using interpolation to improve path planning: The field d* algorithm. Journal of Field Robotics 23(2), pp. 79–101.

[Ferguson et al., 2008] Ferguson, D., Howard, T. M. and Likhachev, M., 2008. Motion planning in urban environments. Journal of Field Robotics 25(11-12), pp. 939–960.

[Ferguson et al., 2004] Ferguson, D., Stentz, A. and Thrun, S., 2004. Pao for planning with hidden state. In: Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, Vol. 3, pp. 2840–2847 Vol.3.

[Flemisch et al., 2008a] Flemisch, F., Kelsch, J., Löper, C., Schieben, A., Schindler, J. and Heesen, M., 2008a. Cooperative control and active interfaces for vehicle assitsance and automation.

[Flemisch et al., 2010] Flemisch, F., Nashashibi, F., Rauch, N., Schieben, A., Glaser, S., Temme, G., Resende, P., Vanholme, B., Löper, C., Thomaidis, G. et al., 2010. Towards highly automated driving: Intermediate report on the haveit-joint system. In: 3rd European Road Transport Research Arena, TRA 2010.

[Flemisch et al., 2003] Flemisch, F. O., Adams, C. A., Conway, S. R., Goodrich, K. H., Palmer, M. T. and Schutte, P. C., 2003. The h-metaphor as a guideline for vehicle automation and interaction.

[Flemisch et al., 2014] Flemisch, F. O., Bengler, K., Bubb, H., Winner, H. and Bruder, R., 2014. Towards cooperative guidance and control of highly automated vehicles: H-mode and conduct-by-wire. Ergonomics 57(3), pp. 343–360.

[Flemisch et al., 2008b] Flemisch, F., Schieben, A., Kelsch, J. and Löper, C., 2008b. Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation. Human Factors for assistance and automation.

[Fletcher et al., 2008] Fletcher, L., Teller, S., Olson, E., Moore, D., Kuwata, Y., How, J., Leonard, J., Miller, I., Campbell, M., Huttenlocher, D. et al., 2008. The mit–cornell collision and why it happened. Journal of Field Robotics 25(10), pp. 775–807.

[Fraichard and Scheuer, 2004] Fraichard, T. and Scheuer, A., 2004. From reeds and shepp's to continuous-curvature paths. Robotics, IEEE Transactions on 20(6), pp. 1025–1035.

[Fuji et al., 2014] Fuji, H., Xiang, J., Tazaki, Y., Levedahl, B. and Suzuki, T., 2014. Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, pp. 407–413.

[Fulgenzi et al., 2009] Fulgenzi, C., Spalanzani, A. and Laugier, C., 2009. Probabilistic motion planning among moving obstacles following typical motion patterns. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, pp. 4027–4033.

[Fulgenzi et al., 2008] Fulgenzi, C., Tay, C., Spalanzani, A. and Laugier, C., 2008. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, pp. 1056–1062.

[Funke and Gerdes, 2016] Funke, J. and Gerdes, J. C., 2016. Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints. Journal of Dynamic Systems, Measurement, and Control 138(2), pp. 021002.

[Funke et al., 2012] Funke, J., Theodosis, P., Hindiyeh, R., Stanek, G., Kritatakirana, K., Gerdes, C., Langer, D., Hernandez, M., Muller-Bessler, B. and Huhnke, B., 2012. Up to the limits: Autonomous audi tts. In: Intelligent Vehicles Symposium (IV), 2012 IEEE, pp. 541–547.

[Furgale et al., 2013] Furgale, P., Schwesinger, U., Rufli, M., Derendarz, W., Grimmett, H., Muhlfellner, P., Wonneberger, S., Timpner, J., Rottmann, S., Li, B., Schmidt, B., Nguyen, T., Cardarelli, E., Cattani, S., Bruning, S., Horstmann, S., Stellmacher, M., Mielenz, H., Koser, K., Beermann, M., Hane, C., Heng, L., Lee, G. H., Fraundorfer, F., Iser, R., Triebel, R., Posner, I., Newman, P., Wolf, L., Pollefeys, M., Brosig, S., Effertz, J., Pradalier, C. and Siegwart,

R., 2013. Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, pp. 809–816.

[Gandhi and Trivedi, 2007] Gandhi, T. and Trivedi, M., 2007. Pedestrian protection systems: Issues, survey, and challenges. Intelligent Transportation Systems, IEEE Transactions on 8(3), pp. 413–430.

[Gasser and Westhoff, 2012] Gasser, T. M. and Westhoff, D., 2012. Bast-study: Definitions of automation and legal issues in germany. In: Proceedings of the 2012 Road Vehicle Automation Workshop.

[Geiger et al., 2012] Geiger, A., Lauer, M., Moosmann, F., Ranft, B., Rapp, H., Stiller, C. and Ziegler, J., 2012. Team annieway's entry to the 2011 grand cooperative driving challenge. Intelligent Transportation Systems, IEEE Transactions on 13(3), pp. 1008–1017.

[Glaser et al., 2010] Glaser, S., Vanholme, B., Mammar, S., Gruyer, D. and Nouveliere, L., 2010. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. Intelligent Transportation Systems, IEEE Transactions on 11(3), pp. 589–606.

[González and Perez, 2013] González, D. and Perez, J., 2013. Control architecture for cybernetic transportation systems in urban environments. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 1119–1124.

[González et al., 2016a] González, D., Milanés, V., Pérez, J. and Nashashibi, F., 2016a. Speed profile generation based on quintic bézier curves for enhanced passenger comfort. In: Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on, IEEE, pp. 814–819.

[González et al., 2017] González, D., Pérez, J. and Milanés, V., 2017. Parametric-based path generation for automated vehicles at roundabouts. Expert Systems with Applications 71, pp. 332–341.

[Gonzalez et al., 2014] Gonzalez, D., Perez, J., Lattarulo, R., Milanes, V. and Nashashibi, F., 2014. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. In: Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, pp. 1430–1435.

[González et al., 2016b] González, D., Pérez, J., Milanés, V. and Nashashibi, F., 2016b. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems 17(4), pp. 1135–1145.

[Griffiths and Gillespie, 2005] Griffiths, P. G. and Gillespie, R. B., 2005. Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits. Human Factors: The Journal of the Human Factors and Ergonomics Society 47(3), pp. 574–590.

[Gu and Dolan, 2012] Gu, T. and Dolan, J. M., 2012. On-road motion planning for autonomous vehicles. In: Intelligent Robotics and Applications, Springer, pp. 588–597.

[Gu and Dolan, 2014] Gu, T. and Dolan, J. M., 2014. Toward human-like motion planning in urban environments. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, pp. 350–355.

[Gu et al., 2013] Gu, T., Snider, J., Dolan, J. and woo Lee, J., 2013. Focused trajectory planning for autonomous on-road driving. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, pp. 547–552.

[Hacker, 1962] Hacker, R., 1962. Certification of algorithm 112: position of point relative to polygon. Communications of the ACM 5(12), pp. 606.

[Han et al., 2010] Han, L., Yashiro, H., Nejad, H. T. N., Do, Q. H. and Mita, S., 2010. Bezier curve based path planning for autonomous vehicle in urban environment. In: Intelligent Vehicles Symposium (IV), 2010 IEEE, IEEE, pp. 1036–1042.

[Han et al., 2008] Han, S., Choi, B. and Lee, J., 2008. A precise curved motion planning for a differential driving mobile robot. Mechatronics 18(9), pp. 486 – 494.

[Hardy and Campbell, 2013] Hardy, J. and Campbell, M., 2013. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. Robotics, IEEE Transactions on 29(4), pp. 913–929.

[Haschke et al., 2008] Haschke, R., Weitnauer, E. and Ritter, H., 2008. On-line planning of time-optimal, jerk-limited trajectories. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, pp. 3248–3253.

[Havlak and Campbell, 2014] Havlak, F. and Campbell, M., 2014. Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. Robotics, IEEE Transactions on 30(2), pp. 461–474.

[Hesse et al., 2011] Hesse, T., Engström, J., Johansson, E., Varalda, G., Brockmann, M., Ramballdini, A., Fricke, N., Flemisch, F., Köster, F. and Kanstrup, L., 2011. Towards user-centred development of integrated information, warning, and intervention strategies for multiple adas in the eu project interactive. In: International Conference on Universal Access in Human-Computer Interaction, Springer, pp. 280–289.

[Hoc, 2001] Hoc, J.-M., 2001. Towards a cognitive approach to human–machine cooperation in dynamic situations. International journal of human-computer studies 54(4), pp. 509–540.

[Horst and Barbera, 2006] Horst, J. and Barbera, A., 2006. Trajectory generation for an on-road autonomous vehicle. In: Defense and Security Symposium, International Society for Optics and Photonics, pp. 62302J–62302J.

[Howard and Kelly, 2007] Howard, T. M. and Kelly, A., 2007. Optimal rough terrain trajectory generation for wheeled mobile robots. The International Journal of Robotics Research 26(2), pp. 141–166.

[Howard et al., 2008] Howard, T. M., Green, C. J., Kelly, A. and Ferguson, D., 2008. State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. Journal of Field Robotics 25(6-7), pp. 325–345.

[Hsieh and Ozguner, 2008] Hsieh, M. F. and Ozguner, U., 2008. A parking algorithm for an autonomous vehicle. In: Intelligent Vehicles Symposium, 2008 IEEE, pp. 1155–1160.

[Hu and Zheng, 2009] Hu, S. and Zheng, G., 2009. Driver drowsiness detection with eyelid related parameters by support vector machine. Expert Systems with Applications 36(4), pp. 7651–7658.

[hua Hsu et al., 2008] hua Hsu, T., Liu, J.-F., Yu, P.-N., Lee, W.-S. and Hsu, J.-S., 2008. Development of an automatic parking system for vehicle. In: Proc. IEEE Vehicle Power and Propulsion Conf. VPPC '08, pp. 1–6.

[Hwang et al., 2003] Hwang, J. Y., Kim, J. S., Lim, S. S. and Park, K. H., 2003. A fast path planning by path graph optimization. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 33(1), pp. 121–129.

[Hwang and Ahuja, 1992] Hwang, Y. K. and Ahuja, N., 1992. Gross motion planning—a survey. ACM Computing Surveys (CSUR) 24(3), pp. 219–291.

[Ishida and Gayko, 2004] Ishida, S. and Gayko, J. E., 2004. Development, evaluation and introduction of a lane keeping assistance system. In: Intelligent Vehicles Symposium, 2004 IEEE, IEEE, pp. 943–944.

[ISO, 1997] ISO, 1997. Mechanical vibration and shock-evaluation of human exposure to whole-body vibration-part 1: General requirements. Technical Report ISO 2631-1, International Organization for Standardization.

[Jeon et al., 2013] Jeon, J. H., Cowlagi, R., Peters, S., Karaman, S., Frazzoli, E., Tsiotras, P. and Iagnemma, K., 2013. Optimal motion planning with the half-car dynamical model for autonomous high-speed driving. In: American Control Conference (ACC), 2013, pp. 188–193.

[Jo et al., 2014] Jo, K., Kim, J., Kim, D., Jang, C. and Sunwoo, M., 2014. Development of autonomous car–part i: Distributed system architecture and development process. Industrial Electronics, IEEE Transactions on.

[Jo et al., 2015] Jo, K., Kim, J., Kim, D., Jang, C. and Sunwoo, M., 2015. Development of autonomous car–part ii: A case study on the implementation of an autonomous driving system based on distributed architecture. Industrial Electronics, IEEE Transactions on.

[Jo et al., 2013] Jo, K., Lee, M., Kim, D., Kim, J., Jang, C., Kim, E., Kim, S., Lee, D., Kim, C., Kim, S. et al., 2013. Overall reviews of autonomous vehicle a1-system architecture and algorithms. In: Intelligent Autonomous Vehicles, Vol. 8, pp. 114–119.

[Kala and Warwick, 2011] Kala, R. and Warwick, K., 2011. Planning of multiple autonomous vehicles using rrt. In: Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on, pp. 20–25.

[Kala and Warwick, 2013] Kala, R. and Warwick, K., 2013. Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization. Journal of Intelligent & Robotic Systems 72(3-4), pp. 559–590.

[Kammel et al., 2008] Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebl, M., Hundelshausen, F. v. et al., 2008. Team annieway's autonomous system for the 2007 darpa urban challenge. Journal of Field Robotics 25(9), pp. 615–639.

[Karaman and Frazzoli, 2010] Karaman, S. and Frazzoli, E., 2010. Optimal kinodynamic motion planning using incremental sampling-based methods. In: Decision and Control (CDC), 2010 49th IEEE Conference on, pp. 7681–7687.

[Karaman and Frazzoli, 2011] Karaman, S. and Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research 30(7), pp. 846–894.

[Karaman et al., 2011] Karaman, S., Walter, M. R., Perez, A., Frazzoli, E. and Teller, S., 2011. Anytime motion planning using the rrt*. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, pp. 1478–1483.

[Karush et al., 2016] Karush, S., Ewens, S. and Stewart, K., 2016. Driver seat: Robot cars won't retire crashtest dummies anytime soon. Technical report, Insurance institute for Highway Safety and Highway Loss Data Institute.

[Kavraki et al., 1996] Kavraki, L. E., Svestka, P., Latombe, J.-C. and Overmars, M. H., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Robotics and Automation, IEEE Transactions on 12(4), pp. 566–580.

[Keller et al., 2011] Keller, C. G., Dang, T., Fritz, H., Joos, A., Rabe, C. and Gavrila, D. M., 2011. Active pedestrian safety by automatic braking and evasive steering. Intelligent Transportation Systems, IEEE Transactions on 12(4), pp. 1292–1304.

[Kelly et al., 2006] Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T. et al., 2006. Toward reliable off road autonomous vehicles operating in challenging environments. The International Journal of Robotics Research 25(5-6), pp. 449–483.

[Kogan and Murray, 2006] Kogan, D. and Murray, R., 2006. Optimization-based navigation for the darpa grand challenge. In: Conference on Decision and Control (CDC).

[Kritayakirana and Gerdes, 2012] Kritayakirana, K. and Gerdes, J. C., 2012. Autonomous vehicle control at the limits of handling. International Journal of Vehicle Autonomous Systems 10(4), pp. 271–296.

[Krotkov et al., 2007] Krotkov, E., Fish, S., Jackel, L., McBride, B., Perschbacher, M. and Pippine, J., 2007. The darpa perceptor evaluation experiments. Autonomous Robots 22(1), pp. 19–35.

[Kummerle et al., 2009] Kummerle, R., Hahnel, D., Dolgov, D., Thrun, S. and Burgard, W., 2009. Autonomous driving in a multi-level parking structure. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 3395–3400.

[Kunchev et al., 2006] Kunchev, V., Jain, L., Ivancevic, V. and Finn, A., 2006. Path planning and obstacle avoidance for autonomous mobile robots: A review. In: Knowledge-Based Intelligent Information and Engineering Systems, Springer, pp. 537–544.

[Kushleyev and Likhachev, 2009] Kushleyev, A. and Likhachev, M., 2009. Time-bounded lattice for efficient planning in dynamic environments. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 1662–1668.

[Kutila et al., 2014] Kutila, M., Pyykönen, P., van Koningsbruggen, P., Pallaro, N. and Pérez-Rastelli, J., 2014. The deserve project: Towards future adas functions. In: Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014 International Conference on, IEEE, pp. 308–313.

[Kuwata et al., 2008] Kuwata, Y., Fiore, G. A., Teo, J., Frazzoli, E. and How, J. P., 2008. Motion planning for urban driving using rrt. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, pp. 1681–1686.

[Kuwata et al., 2009] Kuwata, Y., Karaman, S., Teo, J., Frazzoli, E., How, J. and Fiore, G., 2009. Real-time motion planning with applications to autonomous urban driving. Control Systems Technology, IEEE Transactions on 17(5), pp. 1105–1118.

[Labakhua et al., 2008] Labakhua, L., Nunes, U., Rodrigues, R. and Leite, F. S., 2008. Smooth trajectory planning for fully automated passengers vehicles: spline and clothoid based methods and its simulation. In: Informatics in Control Automation and Robotics, Springer, pp. 169–182.

[LaValle and Hutchinson, 1998] LaValle, S. and Hutchinson, S., 1998. Optimal motion planning for multiple robots having independent goals. Robotics and Automation, IEEE Transactions on 14(6), pp. 912–925.

[LaValle, 2006] LaValle, S. M., 2006. Planning algorithms. Cambridge university press.

[LaValle and Kuffner, 2001] LaValle, S. M. and Kuffner, J. J., 2001. Randomized kinodynamic planning. The International Journal of Robotics Research 20(5), pp. 378–400.

[Lee and Park, 2012] Lee, J. and Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. Intelligent Transportation Systems, IEEE Transactions on 13(1), pp. 81–90.

[Lee and Yu, 2012] Lee, S. and Yu, Y., 2012. Study of the night vision system in vehicle. In: 2012 IEEE Vehicle Power and Propulsion Conference.

[Lee et al., 2011] Lee, S. J., Jo, J., Jung, H. G., Park, K. R. and Kim, J., 2011. Real-time gaze estimator based on driver's head orientation for forward collision warning system. IEEE Transactions on Intelligent Transportation Systems 12(1), pp. 254–267.

[Levinson et al., 2011] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V. et al., 2011. Towards fully autonomous driving: Systems and algorithms. In: Intelligent Vehicles Symposium (IV), 2011 IEEE, IEEE, pp. 163–168.

[Li et al., 2012] Li, L., Wen, D., Zheng, N.-N. and Shen, L.-C., 2012. Cognitive cars: A new frontier for adas research. Intelligent Transportation Systems, IEEE Transactions on 13(1), pp. 395–407.

[Li et al., 2009] Li, Q., Zeng, Z., Yang, B. and Zhang, T., 2009. Hierarchical route planning based on taxi gps-trajectories. In: Geoinformatics, 2009 17th International Conference on, pp. 1–5.

[Liang et al., 2012] Liang, Z., Zheng, G. and Li, J., 2012. Automatic parking path optimization based on bezier curve fitting. In: Automation and Logistics (ICAL), 2012 IEEE International Conference on, pp. 583–587.

[Likhachev and Ferguson, 2009] Likhachev, M. and Ferguson, D., 2009. Planning long dynamically feasible maneuvers for autonomous vehicles. The International Journal of Robotics Research 28(8), pp. 933–945.

[Likhachev et al., 2008] Likhachev, M., Ferguson, D., Gordon, G., Stentz, A. and Thrun, S., 2008. Anytime search in dynamic graphs. Artificial Intelligence 172(14), pp. 1613–1643.

[Lin and Ting, 2007] Lin, J.-S. and Ting, W.-E., 2007. Nonlinear control design of anti-lock braking systems with assistance of active suspension. Control Theory Applications, IET 1(1), pp. 343–348.

[Liu, 2002] Liu, S., 2002. An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In: Advanced Motion Control, 2002. 7th International Workshop on, IEEE, pp. 365–370.

[Llorca et al., 2011] Llorca, D. F., Milanés, V., Alonso, I. P., Gavilán, M., Daza, I. G., Pérez, J. and Sotelo, M. A., 2011. Autonomous pedestrian collision avoidance using a fuzzy steering controller. Intelligent Transportation Systems, IEEE Transactions on 12(2), pp. 390–401.

[Loewenau et al., 2006] Loewenau, J., Gresser, K., Wisselmann, D., Richter, W., Rabel, D. and Durekovic, S., 2006. Dynamic pass prediction—a new driver assistance system for superior and safe overtaking. In: Advanced Microsystems for Automotive Applications 2006, Springer, pp. 67–77.

[Luo et al., 2010] Luo, Y., Remillard, J. and Hoetzer, D., 2010. Pedestrian detection in near-infrared night vision system. In: Intelligent Vehicles Symposium (IV), 2010 IEEE, IEEE, pp. 51–58.

[Madas et al., 2013] Madas, D., Nosratinia, M., Keshavarz, M., Sundstrom, P., Philippsen, R., Eidehall, A. and Dahlen, K.-M., 2013. On path planning methods for automotive collision avoidance. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 931–937.

[Malone et al., 2002] Malone, K., van der Wiel, J. and Saugy, B., 2002. Cybernetic transport systems: lessons to be learned from user needs analysis and field experience. In: Intelligent Vehicle Symposium, 2002. IEEE, Vol. 2, pp. 551–556 vol.2.

[Mamdani, 1974] Mamdani, E. H., 1974. Application of fuzzy algorithms for control of simple dynamic plant. In: Proceedings of the Institution of Electrical Engineers, Vol. 121number 12, IET, pp. 1585–1588.

[Mammar et al., 2006] Mammar, S., Glaser, S. and Netto, M., 2006. Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting. IEEE Transactions on Intelligent Transportation Systems 7(2), pp. 226–241.

[Manage et al., 2003] Manage, S., Nakamura, H. and Suzuki, K., 2003. Performance an of roundabouts as an alternative for intersection control in japan. Journal of the Eastern Asia Society for Transportation Studies 5, pp. 871–883.

[Marchal-Crespo et al., 2010] Marchal-Crespo, L., McHughen, S., Cramer, S. C. and Reinkensmeyer, D. J., 2010. The effect of haptic guidance, aging, and initial skill level on motor learning of a steering task. Experimental brain research 201(2), pp. 209–220.

[Marchese, 2006] Marchese, F., 2006. Multiple mobile robots path-planning with mca. In: Autonomic and Autonomous Systems, 2006. ICAS '06. 2006 International Conference on, pp. 56–56.

[McNaughton et al., 2011] McNaughton, M., Urmson, C., Dolan, J. and Lee, J.-W., 2011. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 4889–4895.

[Merdrignac et al., 2015] Merdrignac, P., Pollard, E. and Nashashibi, F., 2015. 2d laser based road obstacle classification for road safety improvement. In: 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO), IEEE, pp. 1–6.

[Milanés et al., 2011] Milanés, V., Alonso, J., Bouraoui, L. and Ploeg, J., 2011. Cooperative maneuvering in close environments among cybercars and dual-mode cars. IEEE Transactions on Intelligent Transportation Systems 12(1), pp. 15–24.

[Milanés et al., 2012a] Milanés, V., Llorca, D. F., Villagrá, J., Pérez, J., Fernández, C., Parra, I., González, C. and Sotelo, M. A., 2012a. Intelligent automatic overtaking system using vision for vehicle detection. Expert Systems with Applications 39(3), pp. 3362–3373.

[Milanés et al., 2014a] Milanés, V., Marouf, M., Perez, J., González, D. and Nashashibi, F., 2014a. Low-speed cooperative car-following fuzzy controller for cybernetic transport systems. In: Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, IEEE, pp. 2075–2080.

[Milanés et al., 2014b] Milanés, V., Shladover, S., Spring, J., Nowakowski, C., Kawazoe, H. and Nakamura, M., 2014b. Cooperative adaptive cruise control in real traffic situations. Intelligent Transportation Systems, IEEE Transactions on 15(1), pp. 296–305.

[Milanés et al., 2012b] Milanés, V., Villagra, J., Godoy, J., Simó, J., Pérez, J. and Onieva, E., 2012b. An intelligent v2i-based traffic management system. IEEE Transactions on Intelligent Transportation Systems 13(1), pp. 49–58.

[Miller et al., 2008] Miller, I., Campbell, M., Huttenlocher, D., Kline, F.-R., Nathan, A., Lupashin, S., Catlin, J., Schimpf, B., Moran, P., Zych, N., Garcia, E., Kurdziel, M. and Fujishima, H., 2008. Team cornell's skynet: Robust perception and planning in an urban environment. Journal of Field Robotics 25(8), pp. 493–527.

[Molinete et al., 2009] Molinete, B., Bouraui, L., Naranjo, J., Kostense, H., Hendriks, J., Alonso, J., Lobrino, R. and Isasi, L., 2009. Cybercars-2: Close communications for cooperation between cybercars. Technical report, Technical Report Project No IST-2004-0228062.

[Montemerlo et al., 2008] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B. et al., 2008. Junior: The stanford entry in the urban challenge. Journal of field Robotics 25(9), pp. 569–597.

[Montes et al., 2008] Montes, N., Herraez, A., Armesto, L. and Tornero, J., 2008. Real-time clothoid approximation by rational bezier curves. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 2246–2251.

[Montes et al., 2007] Montes, N., Mora, M. and Tornero, J., 2007. Trajectory generation based on rational bezier curves as clothoids. In: Intelligent Vehicles Symposium, 2007 IEEE, pp. 505–510.

[Muffert et al., 2012] Muffert, M., Milbich, T., Pfeiffer, D. and Franke, U., 2012. May i enter the roundabout? a time-to-contact computation based on stereo-vision. In: Intelligent Vehicles Symposium (IV), 2012 IEEE, pp. 565–570.

[Mulder et al., 2008] Mulder, M., Abbink, D. A. and Boer, E. R., 2008. The effect of haptic guidance on curve negotiation behavior of young, experienced drivers. In: Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on, IEEE, pp. 804–809.

[Naranjo et al., 2008] Naranjo, J. E., Gonzalez, C., Garcia, R. and De Pedro, T., 2008. Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. IEEE Transactions on Intelligent Transportation Systems 9(3), pp. 438–450.

[Nash et al., 2007] Nash, A., Daniel, K., Koenig, S. and Felner, A., 2007. Theta^*: Any-angle path planning on grids. In: Proceedings of the National Conference on Artificial Intelligence, Vol. 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1177.

[NHTSA, 2013] NHTSA, N. H. T. S. A., 2013. Preliminary statement of policy concerning automated vehicles. Washington, DC pp. 1–14.

[NHTSA, 2014] NHTSA, N. H. T. S. A., 2014. Traffic safety facts: 2012 data. Washington, DC: US Department of Transportation, National Highway Traffic Safety Administration [online].

[NHTSA, 2016] NHTSA, N. H. T. S. A., 2016. 2015 motor vehicle crashes: overview. (Traffic Safety Facts Research Note. Report No. DOT HS 812 318). Washington, DC: National Highway Traffic Safety Administration. 2016, pp. 1–9.

[OMalley et al., 2011] OMalley, R., Glavin, M. and Jones, E., 2011. Vision-based detection and tracking of vehicles to the rear with perspective correction in low-light conditions. IET Intelligent Transport Systems 5(1), pp. 1–10.

[Owsley and McGwin Jr., 2010] Owsley, C. and McGwin Jr., G., 2010. Vision and driving. Vision Research 50(23), pp. 2348 – 2361. Vision Research Reviews.

[Paden et al., 2016] Paden, B., Cap, M., Yong, S. Z., Yershov, D. and Frazzoli, E., 2016. A survey of motion planning and control techniques for self-driving urban vehicles. arXiv preprint arXiv:1604.07446.

[Parasuraman et al., 2000] Parasuraman, R., Sheridan, T. B. and Wickens, C. D., 2000. A model for types and levels of human interaction with automation. IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans 30(3), pp. 286–297.

[Parent, 2007] Parent, M., 2007. Advanced urban transport: Automation is on the way. Intelligent Systems, IEEE 22(2), pp. 9–11.

[Patz et al., 2008] Patz, B. J., Papelis, Y., Pillat, R., Stein, G. and Harper, D., 2008. A practical approach to robotic design for the darpa urban challenge. Journal of Field Robotics 25(8), pp. 528–566.

[Pérez et al., 2013] Pérez, J., Godoy, J., Villagra, J. and Onieva, E., 2013. Trajectory generator for autonomous vehicles in urban environments. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 409 –414.

[Perez et al., 2014] Perez, J., Lattarulo, R. and Nashashibi, F., 2014. Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, pp. 510–515.

[Pérez et al., 2011] Pérez, J., Milanés, V., de Pedro, T. and Vlacic, L., 2011. Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts. In: Proceedings of the 18th World Congress The International Federation of Automatic Control.

[Petermeijer et al., 2015] Petermeijer, S. M., Abbink, D. A. and de Winter, J. C., 2015. Should drivers be operating within an automation-free bandwidth? evaluating haptic steering support systems with different levels of authority. Human Factors: The Journal of the Human Factors and Ergonomics Society 57(1), pp. 5–20.

[Petrov and Nashashibi, 2013] Petrov, P. and Nashashibi, F., 2013. Adaptive steering control for autonomous lane change maneuver. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 835–840.

[Petrov and Nashashibi, 2014] Petrov, P. and Nashashibi, F., 2014. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. Intelligent Transportation Systems, IEEE Transactions on 15(4), pp. 1643–1656.

[Piazzi et al., 2002] Piazzi, A., Lo Bianco, C. G., Bertozzi, M., Fascioli, A. and Broggi, A., 2002. Quintic g2-splines for the iterative steering of vision-based autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems 3, pp. 27 – 36.

[Pivtoraiko and Kelly, 2005] Pivtoraiko, M. and Kelly, A., 2005. Efficient constrained path planning via search in state lattices. In: International Symposium on Artificial Intelligence, Robotics, and Automation in Space.

[Pivtoraiko et al., 2009] Pivtoraiko, M., Knepper, R. A. and Kelly, A., 2009. Differentially constrained mobile robot motion planning in state lattices. Journal of Field Robotics 26(3), pp. 308–333.

[Rajamani, 2011] Rajamani, R., 2011. Vehicle dynamics and control. Springer Science & Business Media.

[Reeds and Shepp, 1990] Reeds, J. and Shepp, L., 1990. Optimal paths for a car that goes both forwards and backwards. Pacific Journal of Mathematics 145(2), pp. 367–393.

[Ren et al., 2006] Ren, J., McIsaac, K. and Patel, R., 2006. Modified newton's method applied to potential field-based navigation for mobile robots. Robotics, IEEE Transactions on 22(2), pp. 384–391.

[Rice, 2010a] Rice, E., 2010a. Mini-roundabouts. Technical report, U.S. Departmet of Trasportation Federal Highway Administration (FHWA-SA-10-007).

[Rice, 2010b] Rice, E., 2010b. Roundabouts. Technical report, U.S. Departmet of Trasportation Federal Highway Administration (FHWA-SA-10-006).

[Robinson et al., 2000] Robinson, B. W., Rodegerdts, L., Scarbrough, W., Kittelson, W., Troutbeck, R., Brilon, W., Bondzio, L., Courage, K., Kyte, M., Mason, J., Flannery, A., Myers, E., Bunker, J. and G., J., 2000. Roundabouts: An informational guide. Technical report, Report FHWA-RD-00-067. FHWA, U.S. Department of Transportation.

[Rodegerdts et al., 2010] Rodegerdts, L., Bansen, J., Tiesler, C., Knudsen, J. and Myers, E., 2010. Roundabouts: An informational guide. Technical report, National Cooperative Highway Research Program.

[Roldao et al., 2015] Roldao, L., Perez Rastelli, J., González, D. and Milanés, V., 2015. Description and technical specification of cybernetic transportation systems: an urban transportation concept -citymobil2 project approach-. In: IEEE ITSC, pp. 176–181.

[Romani and Sabin, 2004] Romani, L. and Sabin, M., 2004. The conversion matrix between uniform b-spline and bézier representations. Computer Aided Geometric Design 21(6), pp. 549 – 560.

[Rufli et al., 2009] Rufli, M., Siegwart, R., Siegwart, R. Y. and Siegwart, R. Y., 2009. On the application of the d* search algorithm to time-based planning on lattice graphs. ECMR 9, pp. 105–110.

[Ryu et al., 2013] Ryu, J.-H., Ogay, D., Bulavintsev, S., Kim, H. and Park, J.-S., 2013. Development and experiences of an autonomous vehicle for high-speed navigation and obstacle avoidance. In: Frontiers of Intelligent Autonomous Systems, Springer, pp. 105–116.

[SAE, 2014] SAE, O.-R. A. V. S. C., 2014. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems.

[Sahayadhas et al., 2012] Sahayadhas, A., Sundaraj, K. and Murugappan, M., 2012. Detecting driver drowsiness based on sensors: a review. Sensors 12(12), pp. 16937–16953.

[Saleh et al., 2013] Saleh, L., Chevrel, P., Claveau, F., Lafay, J.-F. and Mars, F., 2013. Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty. Intelligent Transportation Systems, IEEE Transactions on 14(2), pp. 974–983.

[Sánchez-Reyes and Chacón, 2003] Sánchez-Reyes, J. and Chacón, J., 2003. Polynomial approximation to clothoids via s-power series. Computer-Aided Design 35(14), pp. 1305 – 1313.

[Schmidt and Wulf, 1997] Schmidt, R. A. and Wulf, G., 1997. Continuous concurrent feedback degrades skill learning: Implications for training and simulation. Human factors 39(4), pp. 509.

[Shamir, 2004] Shamir, T., 2004. How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory. Automatic Control, IEEE Transactions on 49(4), pp. 607–610.

[Sheridan and Verplank, 1978] Sheridan, T. B. and Verplank, W. L., 1978. Human and computer control of undersea teleoperators. Technical report, DTIC Document.

[Shiller and Gwo, 1991] Shiller, Z. and Gwo, Y.-R., 1991. Dynamic motion planning of autonomous vehicles. Robotics and Automation, IEEE Transactions on 7(2), pp. 241–249.

[Shimrat, 1962] Shimrat, M., 1962. Algorithm 112: position of point relative to polygon. Communications of the ACM 5(8), pp. 434.

[Shladover, 1997] Shladover, S., 1997. The gm-path platoon scenario. Intellimotion. Vol. 6, no. 3.

[Shladover et al., 1991] Shladover, S., Desoer, C. A., Hedrick, J. K., Tomizuka, M., Walrand, J., Zhang, W.-B., McMahon, D. H., Peng, H., Sheikholeslam, S. and McKeown, N., 1991. Automated vehicle control developments in the path program. Vehicular Technology, IEEE Transactions on 40(1), pp. 114–130.

[Shladover, 2007] Shladover, S. E., 2007. Path at 20-history and major milestones. IEEE Transactions on intelligent transportation systems 8(4), pp. 584–592.

[Simon and Becker, 1999] Simon, A. and Becker, J., 1999. Vehicle guidance for an autonomous vehicle. In: Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on, pp. 429–434.

[Smith, 2012] Smith, B. W., 2012. Automated vehicles are probably legal in the united states. Center for Internet and Society.

[Stentz, 1994] Stentz, A., 1994. Optimal and efficient path planning for partially-known environments. In: Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, IEEE, pp. 3310–3317.

[Thorpe et al., 1998] Thorpe, C., Jochem, T. and Pomerleau, D., 1998. Automated highways and the free agent demonstration. In: Robotics Research, Springer, pp. 246–254.

[Thrun et al., 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G. et al., 2006. Stanley: The robot that won the darpa grand challenge. Journal of field Robotics 23(9), pp. 661–692.

[Tracza and Chodura, 2012] Tracza, M. and Chodura, J., 2012. Performance and safety of roundabouts with traffic signals. In: SIIV 5th International Congress Sustainability of Road Infrastructures, pp. 789 – 800.

[Trehard et al., 2014] Trehard, G., Alsayed, Z., Pollard, E., Bradai, B. and Nashashibi, F., 2014. Credibilist simultaneous localization and mapping with a lidar. In: IEEE/RSJ International Conference onIntelligent Robots and Systems (IROS 2014), pp. 2699–2706.

[Trepagnier et al., 2006] Trepagnier, P. G., Nagel, J., Kinney, P. M., Koutsougeras, C. and Dooner, M., 2006. Kat-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain. Journal of Field Robotics 23(8), pp. 509–526.

[Trimble et al., 2014] Trimble, T. E., Bishop, R., Morgan, J. F., Blanco, M. et al., 2014. Human factors evaluation of level 2 and level 3 automated driving concepts: Past research, state of automation technology, and emerging system concepts.

[Urmson et al., 2008] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C. et al., 2008. Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics 25(8), pp. 425–466.

[Urmson et al., 2006] Urmson, C., Ragusa, C., Ray, D., Anhalt, J., Bartz, D., Galatali, T., Gutierrez, A., Johnston, J., Harbaugh, S., Messner, W. et al., 2006. A robust approach to high-speed navigation for unrehearsed desert terrain. Journal of Field Robotics 23(8), pp. 467–508.

[Vaca et al., 2016] Vaca, M., Traver, J. E., Milanés, V., Pérez, J., Bautista, D. G. and Nashashibi, F., 2016. Automated global planner for cybernetic transportation systems. In: 14th international conference on control, automation, robotics and vision (ICARCV).

[van Dijke and van Schijndel, 2012] van Dijke, J. and van Schijndel, M., 2012. Citymobil, advanced transport for the urban environment: Update. Transportation Research Record: Journal of the Transportation Research Board 2324(2324), pp. 29–36.

[Van Leeuwen et al., 2011] Van Leeuwen, P., De Groot, S., Happee, R. and De Winter, J., 2011. Effects of concurrent continuous visual feedback on learning the lane keeping task. In: Proceedings of the 6th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 482–488.

[van Nunen et al., 2012] van Nunen, E., Kwakkernaat, R., Ploeg, J. and Netten, B. D., 2012. Cooperative competition for future mobility. Intelligent Transportation Systems, IEEE Transactions on 13(3), pp. 1018–1025.

[Van Schijndel-de Nooij et al., 2011] Van Schijndel-de Nooij, M., Krosse, B., Van den Broek, T., Maas, S., Van Nunen, E., Zwijnenberg, H., Schieben, A., Mosebach, H., Ford, N., McDonald, M. and Sanchez, J., 2011. Definition of necessary vehicle and infrastructure systems for automated driving. SMART 2010/0064 European Commission.

[Vanholme et al., 2013] Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S. and Mammar, S., 2013. Highly automated driving on highways based on legal safety. IEEE Transactions on Intelligent Transportation Systems 14(1), pp. 333–347.

[Villagra et al., 2012] Villagra, J., Milanés, V., Pérez, J. and Godoy, J., 2012. Smooth path and speed planning for an automated public transport vehicle. Robotics and Autonomous Systems 60(2), pp. 252–265.

[Vorobieva et al., 2014] Vorobieva, H., Glaser, S., Minoiu-Enache, N. and Mammar, S., 2014. Automatic parallel parking with geometric continuous-curvature path planning. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, pp. 465–471.

[Vorobieva et al., 2013] Vorobieva, H., Minoiu-Enache, N., Glaser, S. and Mammar, S., 2013. Geometric continuous-curvature path planning for automatic parallel parking. In: Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on, pp. 418–423.

[Walker Smith, 2013] Walker Smith, B., 2013. Automated driving: Legislative and regulatory action. Technical report, The Center for Internet and Society.

[Walton and Meek, 2005] Walton, D. and Meek, D., 2005. A controlled clothoid spline. Computers & Graphics 29(3), pp. 353 – 363.

[Walton et al., 2003] Walton, D., Meek, D. and Ali, J., 2003. Planar< i> g</i>< sup> 2</sup> transition curves composed of cubic bézier spiral segments. Journal of Computational and Applied Mathematics 157(2), pp. 453–476.

[Wang et al., 2001] Wang, L., Miura, K., Nakamae, E., Yamamoto, T. and Wang, T., 2001. An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves. Computer-Aided Design 33(14), pp. 1049 – 1058.

[Wei et al., 2013] Wei, J., Snider, J. M., Kim, J., Dolan, J. M., Rajkumar, R. and Litkouhi, B., 2013. Towards a viable autonomous driving research platform. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 763–770.

[Werling et al., 2008] Werling, M., Gindele, T., Jagszent, D. and Groll, L., 2008. A robust algorithm for handling moving traffic in urban scenarios. In: Intelligent Vehicles Symposium, 2008 IEEE, IEEE, pp. 1108–1112.

[Werling et al., 2011] Werling, M., Kammel, S., Ziegler, J. and Gröll, L., 2011. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. The International Journal of Robotics Research.

[Werling et al., 2010] Werling, M., Ziegler, J., Kammel, S. and Thrun, S., 2010. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, pp. 987–993.

[woo Lee and Litkouhi, 2012] woo Lee, J. and Litkouhi, B., 2012. A unified framework of the automated lane centering/changing control for motion smoothness adaptation. In: Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, pp. 282–287.

[Xiaoguang et al., 2004] Xiaoguang, Y., Xiugang, L. and Kun, X., 2004. A new traffic-signal control for modern roundabouts method and application. IEEE Intelligent Transportation Systems Society 5, pp. 282 – 287.

[Xie et al., 2010] Xie, J., Nashashibi, F., Parent, M. and Favrot, O. G., 2010. A real-time robust global localization for autonomous mobile robots in large environments. In: 11th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, pp. 1397 – 1402.

[Xu et al., 2014] Xu, W., Pan, J., Wei, J. and Dolan, J., 2014. Motion planning under uncertainty for on-road autonomous driving. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 2507–2512.

[Xu et al., 2012] Xu, W., Wei, J., Dolan, J., Zhao, H. and Zha, H., 2012. A real-time motion planner with trajectory optimization for autonomous vehicles. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 2061–2067.

[Yang and Sukkarieh, 2010] Yang, K. and Sukkarieh, S., 2010. An analytical continuous-curvature path-smoothing algorithm. Robotics, IEEE Transactions on 26(3), pp. 561–568.

[Young et al., 2008] Young, K., Lee, J. D. and Regan, M. A., 2008. Driver distraction: Theory, effects, and mitigation. CRC Press.

[Yu and Qin, 2009] Yu, L. and Qin, C., 2009. Real-time signal control method for multi-approach roundabouts. In: International Conference on Management and Service Science (MASS), pp. 1–4.

[Zhang et al., 2015] Zhang, K., Zhang, D., de La Fortelle, A., Wu, X. and Grégoire, J., 2015. State-driven priority scheduling mechanisms for driverless vehicles approaching intersections. IEEE Transactions on Intelligent Transportation Systems 16(5), pp. 2487–2500.

[Zhang, 2011] Zhang, Y., 2011. Visual and cognitive distraction effects on driver behavior and an approach to distraction state classification. North Carolina State University.

[Ziegler and Stiller, 2009] Ziegler, J. and Stiller, C., 2009. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, pp. 1879–1884.

[Ziegler et al., 2014a] Ziegler, J., Bender, P., Dang, T. and Stiller, C., 2014a. Trajectory planning for bertha—a local, continuous method. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, pp. 450–457.

[Ziegler et al., 2014b] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C., Kaus, E., Herrtwich, R., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M. and Zeeb, E., 2014b. Making bertha drive —an autonomous journey on a historic route. Intelligent Transportation Systems Magazine, IEEE 6(2), pp. 8–20.

[Ziegler et al., 2008] Ziegler, J., Werling, M. and Schroder, J., 2008. Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In: Intelligent Vehicles Symposium, 2008 IEEE, IEEE, pp. 787–791.

Résumé

Un des buts de la recherche et du développement des systèmes de transport intelligents est d'augmenter le confort et la sécurité des passagers, tout en réduisant la consommation d'énergie, la pollution de l'air et le temps de déplacement.

L'introduction de voitures complètement autonomes sur la voie publique nécessite la résolution d'un certain nombre de problèmes techniques et en particulier de disposer de modules de planification de trajectoire robustes. Ce travail de thèse s'inscrit dans ce cadre. Il propose une architecture modulaire pour la planification de trajectoire d'un véhicule autonome.

La méthode permet de générer des trajectoires constituées de courbes interpolées adaptées à des environnements complexes comme des virages, des ronds-points, etc., tout en garantissant la sécurité et le confort des passagers. La prise en compte de l'incertitude des systèmes de perception, des limites physiques du véhicule, de la disposition des routes et des règles de circulation est aussi assurée dans le calcul de la trajectoire. L'algorithme est capable de modifier en temps réel la trajectoire prédéfinie de façon à éviter les collisions. Le calcul de la nouvelle trajectoire maintient les accélérations latérales à leur minimum, assurant ainsi le confort du passager. L'approche proposée a été évaluée et validée dans des environnements simulés et sur des véhicules réels. Cette méthode permet d'éviter les obstacles statiques et dynamiques repérés par le système de perception.

Un système d'aide à la conduite pour le contrôle partagé basé sur cette architecture est introduit. Il prend en compte l'arbitrage, la surveillance et le partage de la conduite tout en maintenant le conducteur dans la boucle de contrôle. Il laisse le conducteur agir tant qu'il n'y a pas de danger et interagit avec le conducteur dans le cas contraire. L'algorithme se décompose donc en deux processus : 1) évaluation du risque et, s'il y a un risque avéré 2) partage du contrôle à l'aide de signaux haptiques via le volant.

La méthode de planification de trajectoire présentée dans cette thèse est modulaire et générique. Elle peut être intégrée facilement dans toute architecture d'un véhicule autonome.

Mots Clés

Planification des trajectoires, contrôle partagé, véhicules autonomes, systèmes intelligent de transport.

Abstract

Developments in the Intelligent Transportation Systems (ITS) field show promising results at increasing passengers comfort and safety, while decreasing energy consumption, emissions and travel time. In road transportation, the appearance of automated vehicles is significantly aiding drivers by reducing some driving-associated tedious tasks. However, there is still a long way to go before making the transition between automated vehicles (i.e. vehicles with some automated features) and autonomous vehicles on public roads (i.e. fully autonomous driving), specially from the motion planning point of view. With this in mind, the present PhD thesis proposes the design of a generic modular architecture for automated vehicles motion planning. It implements and improves curve interpolation techniques in the motion planning literature by including comfort as the main design parameter, addressing complex environments such as turns, intersections and roundabouts. It will be able to generate suitable trajectories that consider measurements' incertitude from the perception system, vehicle's physical limits, the road layout and traffic rules. In case future collision states are detected, the proposed approach is able to change---in real-time---the current trajectory and avoid the obstacle in front. It permits to avoid obstacles in conflict with the current trajectory of the ego-vehicle, considering comfort limits and developing a new trajectory that keeps lateral accelerations at its minimum. The proposed approach is tested in simulated and real urban environments, including turns and two-lane roundabouts with different radii. Static and dynamic obstacles are considered as to face and interact with other road actors, avoiding collisions when detected. The functional architecture is also tested in shared control and arbitration applications, focusing in keeping the driver in the control loop to addition the system's supervision over drivers' knowledge and skills in the driving task. The control sharing advanced driver assistance system (ADAS) is proposed in two steps: 1) risk assessment of the situation in hand, based on the optimal trajectory and driving boundaries identified by the motion planning architecture and; 2) control sharing via haptic signals sent to the driver through the steering wheel. The approach demonstrates the modularity of the functional architecture as it proposes a general solution for some of today's unsolved challenges in the automated driving field.

Keywords

Trajectory planning, shared control, autonomous vehicles, intelligent transportation systems.