université PARIS-SACLAY

NNT : 2019SACLE014

Thèse de doctorat

Motion Planning for Autonomous Highway Driving: A Unified Architecture for Decision-Maker and Trajectory Generator

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université d'Evry-Val-d'Essonne

Ecole doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Automatique

Thèse présentée et soutenue à Versailles, le 27 septembre 2019, par

LAURÈNE CLAUSSMANN

Composition du Jury :

Michel BASSET Professeur, Université de Haute Alsace (ENSISA-IRIMAS)	Rapporteur
Fawzi NASHASHIBI Directeur de Recherche, INRIA (RITS)	Rapporteur
Arnaud DE LA FORTELLE Professeur, MINES ParisTech (CAOR)	Président
Lydie NOUVELIERE Maître de Conférences, UEVE (IBISC)	Examinateur
Sébastien GLASER Professeur, QUT (CARRS-Q)	Directeur de thèse
Olivier ORFILA Chargé de Recherche, IFSTTAR (LIVIC)	Encadrant
Marc REVILLOUD Chercheur, Institut VEDECOM	Encadrant

Abstract

Automation of everyday vehicles becomes a societal, economic and technological issue. This task of driving automation is suggested as one of the major solutions to make transportation safer, more comfortable and potentially more environment friendly. Automated systems are gradually implemented in recent vehicles through Advanced Driver Assistance Systems (ADAS). However, they are often limited to warning systems and a partial delegation of control, or only applied under controlled and constrained conditions.

This thesis work is part of the development of a self-driving car in highway environments. More precisely, it aims to propose a unified architecture of trajectory planner and decision-maker taking into account the limitations of the environment and the available data within the current development of sensors technologies (distance limitations, uncertainties).

On the one hand, the method generates sigmoid trajectories in a continuous spatiotemporal representation of the evolution space, which is reduced beforehand by modeling collision-free intervals in nominal conditions of driving. The sigmoid parameters are subsequently optimized with a simulated annealing approach that uses the decision-maker algorithm as the evaluation function for the generated trajectory. It thus makes it possible to elude both the discretization and position/speed decoupling problems. On the other hand, the aggregation of fuzzy logic and belief theory allows decision making on heterogeneous criteria and uncertain data. The proposed framework also handles personalization of the vehicle's behavior, depending on the passengers' risk perception and an aggressive or conservative driving style.

The presented approach was finally evaluated and validated in a simulation environment, and then in a test vehicle. The planning block was integrated into the existing vehicle's architecture, interfaced with the localization, obstacles' perception and control blocks.

Keywords: Self-driving cars, Motion planning, Decision making, Trajectory generation, Intelligent Transportation Systems

Résumé

L'automatisation des véhicules est un enjeu sociétal, économique et technologique important et d'actualité. Cette tâche de délégation de la conduite est proposée comme l'une des solutions majeures pour rendre les transports plus sûrs, plus confortables et potentiellement plus respectueux de l'environnement. Des systèmes automatisés sont progressivement implantés dans les véhicules récents à travers les systèmes avancés d'aide à la conduite (ADAS). Cependant, ils se limitent essentiellement à une délégation partielle de la conduite, ou à leurs applications dans des conditions contrôlées et contraintes.

Ce travail de thèse s'inscrit dans le développement d'un véhicule autonome en milieu autoroutier. Plus précisément, il s'agit de proposer une architecture unifiée de génération de trajectoires avec une prise de décision prenant en compte les limitations de l'environnement et des informations disponibles actuellement sur un véhicule automatisé.

La méthode propose d'une part de générer des trajectoires sous forme de sigmoïde dans une représentation spatiotemporelle continue de l'espace de navigation, préalablement réduit par la modélisation d'intervalles sans collision en condition nominale de conduite. Les paramètres de la sigmoïde sont ensuite optimisés par une stratégie de recuit simulé utilisant l'algorithme de prise de décision comme fonction d'évaluation de la trajectoire générée. De cette manière, les problèmes de discrétisation et de découplage position/vitesse sont évités. D'autre part, l'agrégation des théories de logique floue et des croyances permet une prise de décision sur des critères hétérogènes et des données incertaines. Le formalisme présenté offre la possibilité d'adapter le comportement du véhicule aux passagers, notamment selon leur perception du risque et leur souhait d'une conduite souple ou sportive.

L'approche développée a finalement été évaluée et validée en environnement de simulation puis sur un véhicule de test. La brique de planification a alors été intégrée à l'architecture existante du véhicule, en aval des briques de localisation et de perception des obstacles et en amont de la brique de contrôle.

Mots clés: Véhicule autonome, Planification de mouvement, Prise de décision multi-critères, Génération de trajectoires, Systèmes de transports intelligents

Remerciements

Je souhaite remercier tout d'abord les professeurs et chercheurs qui ont accepté de participer à mon jury de thèse. Ainsi, je remercie plus particulièrement mes deux rapporteurs, Michel Basset et Fawzi Nashahsibi, dont les retours minutieux et experts m'ont permis de prendre conscience du travail réalisé et d'améliorer la qualité et la clarté du manuscrit et de la soutenance. Merci également à Arnaud de La Fortelle d'avoir accepté la présidence de ce jury, ainsi qu'à Lydie Nouvelière et Olivier Orfila pour les discussions enrichissantes que nous avons pu mener pendant la soutenance, et dont les questions éclairées ont permis de couvrir tous les aspects de mon travail et des perspectives à y apporter. J'ai été sincèrement honorée de votre présence et de l'intérêt que vous avez porté à mes recherches. Je tiens également à remercier Dominique Gruyer d'avoir initié ce projet de thèse et de m'avoir offert la possibilité d'effectuer un séjour au laboratoire ACIS de l'Université de Colombie Britannique, ainsi que Sébastien Glaser pour les discussions autour de la thèse et de m'avoir permis de mener ce projet à son terme.

Je tiens également à témoigner ma profonde gratitude à toute l'équipe VEH08 de l'Institut VEDECOM qui a toujours été présente durant cette thèse et qui m'a toujours soutenue. J'ai eu le plaisir de voir grandir cette équipe depuis mon arrivée, et de découvrir chacun d'entre vous au fil des péripéties de cette thèse. Mes souvenirs avec vous sont nombreux, il y avait toujours quelqu'un pour un sourire, une oreille attentive pour une petite ou une longue discussion, une contrepèterie, des statistiques incongrues, une aide avisée, un croissant et un chocolat chaud, une petite bière et des cacahouètes, un débat sans fin, des encouragements, des relectures, des blagues, des conseils ou encore du réconfort, tout cela avec bienveillance. Merci à (je vous mets par ordre alphabétique pour ne pas faire de jaloux) : Alexis, Ali, Amin, Benoit, Emmanuel, Florent, Francky, Ghina, Hugo, Karima, Lynda, Maryem, Michel, Mohamed, Nihed, Steve, Sylvain, Vincent, Yousri, Yoan. Merci d'avoir cru en moi et de m'avoir fait confiance. Je remercie également tous les collègues de VEDECOM avec qui j'ai pu échanger et qui m'ont toujours tirée vers le haut. Au risque d'en oublier certains (ce dont je m'excuse par avance), un merci particulier à Aziz, Bertrand, Elsa, Floriane, Franck, Guilhem, Katherine, Laurent, Marie, Maxime, Mercedes, Nadim, Nelson, Nicolas, Romy, Sami, Sofiane, Stéphane, avec qui j'ai pu discuter un peu plus dans les couloirs, le bus ou la ligne N. Merci aussi aux fonctions supports et à ceux que j'ai un peu plus embêtés mais qui ont toujours été très efficaces et compréhensifs, Assia, Cédrine, Fethi, Marie, Sabrina, Sarah et Valentin.

Merci aussi aux anciens de l'équipe qui m'ont également apporté leur aide, Fernando, Hatem, J.-C., Lan, Zayed et Zui, et aux anciens collègues de VEDECOM, Mustapha, Céline et Mathilde, qui pourront vous écrire un livre sur mes aventures à force de m'écouter parler.

Pour les plus attentifs, il y a deux personnes que je n'ai pas encore remerciées et qui, je l'espère, seront fières de ce travail de thèse. Guillaume et Marc, vous avez été les piliers de mon épopée et ce manuscrit n'aurait pas été écrit sans vous. Vous ne m'avez jamais laissé tomber et je vous en suis très très sincèrement reconnaissante. Vous consacrer ces quelques lignes me fait monter les larmes aux yeux car, même si je me répète, je ne sais pas comment

vous dire merci pour tout ce que vous avez fait pour moi. Guillaume, tu as su trouver les mots pour me ramener à mon bureau et me redonner confiance quand je pensais avoir tout perdu. Marc, tu as été un véritable compagnon d'aventure. Tu as été l'une des premières personnes avec qui j'ai discuté lors de mon entretien à VEDECOM et tu m'as accompagné jusqu'à la fin. Tu as toujours pris le temps de m'écouter et de me conseiller. Tu m'as apporté ta sincérité et ton respect, qui sont deux valeurs essentielles à mes yeux. Je n'aurais pas pu espérer mieux que tout ce que tu m'as apporté, et tu m'as aussi fait grandir (une parenthèse de merci).

Je n'oublie pas les amis, toujours présents pour se changer les idées et qui malgré mon intermittence, me sont restés fidèles et ont supporté mes mines de déterrée après un rendu d'article. Merci aux copains de Condorcet avec qui nous fêtons aussi nos 10 ans d'amitié, merci aux copains des copains qui sont devenus des copains, les copains de Tournan-en-Brie, ceux des Ponts et ceux de Centrale Paris, merci de m'avoir accueillie ! Un grand merci aussi à ceux dont les sms ont pu rester sans réponse, mais qui ont toujours répondu présents, Quentin et Soline. Et merci à tous ceux que j'ai pu rencontrer lors de ce doctorat en France ou à l'étranger.

Je souhaite également remercier ma famille et ma belle-famille pour leur soutien. Mamy, Papa, Maman et Lili, j'espère que vous avez bien compris ce que je faisais lors de la soutenance car il n'y en aura pas d'autres ! Je souhaite dédier ce travail à mon grand-père qui n'aura pas eu le temps de le voir achevé.

Pour terminer, je remercie de tout cœur Milan. Comme tu le dis dans tes remerciements, « c'était un sacré défi que de commencer nos thèses en ce même jour du 1er décembre 2015 ». Ça l'a été, et pas des moindres. C'était un beau défi à partager avec toi. Tu m'as montré un soutien sans faille, tu as répondu présent pour les discussions, les réflexions, les recherches bibliographiques à 3h du matin quand j'avais un doute existentiel avant d'envoyer mes articles. Merci pour tes relectures, tes encouragements, ton réconfort et ta confiance. Merci d'avoir toujours été là, d'avoir tout partagé avec moi, et je suis prête à relever n'importe quel autre défi à tes côtés.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviation

DST	Dempster-Shafer Theory
FDST	Fuzzy Dempster-Shafer Theory
FIS	Fuzzy Inference System
MCDM	Multi-Criteria Decision Making
NCNI	Non-Collision Nominal Interval
SA	Simulated Annealing

Subscripts and Superscripts

$\square_0$	initial value
$\square_{ego}$	denotes an ego variable
$\square_h$ or ^{h}	denotes the h^{th} hypothesis $\mathcal{H}_h$
$\square_i$	denotes the i^{th} ego profile
$\square_{inf}$ or $\square^{inf}$	inferior bound value
$\square^{\iota}$	denotes the ι^{th} fuzzy set
$\square_j$	denotes the j^{th} obstacle O_j
$\square_k$	denotes the k^{th} interval I_k
$\square_{\kappa}$	denotes the κ^{th} category
$\square^l$	denotes the l^{th} combined output fuzzy set
$\square_{lim}$	limit value for a variable
$\square_m$	denotes the m^{th} maneuver
$\square_{max}$	maximum limit value for a variable
$\square_{min}$	minimum limit value for a variable
$\square_p$	denotes the p^{th} trajectory
$\square_r$	denotes the r^{th} rule
$\square_{ref}$	reference value output of the motion planner
$\square_s$	denotes the s^{th} criterion
$\square_{sup}$ or $\square^{sup}$	superior bound value
$\square_t$	denotes the t^{th} target
$\square_x$	denotes the variable along x-axis (longitudinal)
$\square_y$	denotes the variable along y-axis (lateral)

Frames

$\mathscr{E}$	ego local frame
$\mathscr{O}$	obstacle local frame
$\mathscr{R}$	road local frame
$\mathscr{W}$	world global frame

Functions

$f_h()$	$\mathbb{R} \rightarrow [0,1]$	representation function of hypothesis $\mathcal{H}_h$
$\mu_A()$	$\mathbb{R} \rightarrow [0,1]$	membership function of the fuzzy set A

Operators

$\dot{}$	time derivative
$\ddot{}$	time second derivative
$\Delta()$	difference operator for variables
$end()$	returns the end value of a set
inf()	returns the inferior value of an interval
$isTrue()$	returns the Boolean value ($0 : false$, $1 : true$)
$l()$	returns the length of an interval
max()	returns the maximum value of a set
$mean()$	returns the mean value of a set
min()	returns the minimum value of a set
$sum()$	returns the sum value of a set
sup()	returns the superior value of an interval

Fuzzy Operators

ARR	aggregation function
IRR_r	activation characterization of rule r
ORR_r	implication function of rule r

Vectors

χ	ego state vector	$\chi = (\ddot{x}, \dot{x}, x, \ddot{y}, \dot{y}, y, \alpha)$
χ_{ego}	simple ego state vector	$\chi_{ego} = (x, y, v)$
χ_{O_j}	obstacle j state vector	$\chi_{O_j} = (x, y, v)$
χ_{ref}	reference trajectory over the prediction horizon H_p	$\chi_{ref} = (x(t), y(t), v(t))_{(t=1..H_p)}$
v_{ref}	reference velocity over H_p	$v_{ref} = (v(t))_{(t=1..H_p)}$
ξ	SA variable vector	$\xi = (v_t, c, \lambda)$

Variables

α	ego direction angle with x-axis
a_x	ego longitudinal acceleration
a_y	ego lateral acceleration
b	sigmoid shift parameter
bba	basic belief assignment
bel	belief degree (lower probability)
c	sigmoid delay parameter
d	Euclidean distance
$d_{f,collision}$	front collision distance
$d_{r,collision}$	rear collision distance
$d_{f,safe}$	front safe distance
$d_{r,safe}$	rear safe distance
d_x	longitudinal deceleration
dir	maneuver direction $dir = \{Left, Straight, Right\}$
H_p	prediction horizon
$\mathcal{I}_p$	evaluation index of the p^{th} trajectory
λ	sigmoid parameter
$mf_A(u_s)$	membership degree of the criterion value u_s in the fuzzy set A
pl	plausibility degree (upper probability)
q	geometric parameter of the geometric progression
t	time variable
T	SA temperature
T_0	SA initial temperature
T_f	SA final temperature
u_s	value of criterion s
uod_A	universe of discernment of the fuzzy set A
v	velocity
v_{ego}	ego velocity
v_{max}	maximum road speed limit
v_{min}	minimum road speed limit
v_{O_j}	O_j velocity
v_{rel,O_j}	relative velocity between ego and O_j
v_t	target velocity
v_{wish}	driver's desired velocity
v_x	ego longitudinal velocity
v_y	ego lateral velocity
ν_s	weight of the s^{th} criterion
w_κ	weight of the κ^{th} category
Ω_h	weight of the h^{th} hypothesis
x	position along x-axis in the road $\mathscr{R}$ or ego $\mathscr{E}$ frame
x_{rel,O_j}	relative longitudinal position between ego and O_j
y	position along y-axis in the road $\mathscr{R}$ or ego $\mathscr{E}$ frame
y_{rel,O_j}	relative lateral position between ego and O_j
$y_{centerline}$	lane centerline lateral coordinate in $\mathscr{R}$

Others

$\mathcal{H}$	hypothesis
I	interval
$\mathcal{J}_{obj}$	objective function
LW	local warning risk level (LW_κ^ι for category κ and fuzzy set ι)
O	obstacle
$O^{inf_{I_k}}$	represents the inferior bound of I_k
$O^{sup_{I_k}}$	represents the superior bound of I_k
RA	Risk Assessment

Quantities

C	cardinal of the categories	$\kappa = 1..C$
H	cardinal of the hypotheses $\mathcal{H}$	$h = 1..H$
J	cardinal of the obstacles O	$j = 1..J$
K	cardinal of the intervals I	$k = 1..K$
M	cardinal of the maneuvers	$m = 1..M$
N	cardinal of the ego profiles	$i = 1..N$
n_{LW}	cardinal of the combined output fuzzy sets	$l = 1..n_{LW}$
P	cardinal of the candidate trajectories	$p = 1..P$
S	cardinal of the criteria	$s = 1..S$

Chapter 1

Introduction

Contents

Since the last decade, the development of automated and autonomous vehicles has spread worldwide among universities, software companies and the automotive sector as one of the promising advancements in automotive engineering and research. The automation of everyday vehicles becomes a societal, economic and technological issue. This task of driving automation is studied in the field of transportation safety, comfort and environmental care. Automated systems are gradually implemented in recent vehicles through Advanced Driver Assistance Systems (ADAS).

This first chapter presents the evolution of the context of autonomous vehicles in section 1.1, the motivations driving this doctoral work in section 1.2 and the contributions made in section 1.3. The structure of the document is finally summarized in section 1.4.

1.1 Context of the autonomous vehicle

The contributions to Intelligent Transportation Systems (ITS) aim to provide innovative transport services by significantly tackling safety and security of road transport, including drivers, public passengers and freight mobility, transport networks, as well as environmental performance and efficiency. These contributions handle the development of safer, cleaner, more comfortable and more efficient mobility of people and goods, saving lives, time and money. In [EPC], ITS are defined as systems integrating "telecommunications, electronics and information technologies with transport engineering in order to plan, design, operate, maintain and manage transport systems." ITS thus involve a wide range of applications, from the traffic management to the autonomous vehicles, which is the scope of this work.

This section overviews the main research projects on the autonomous vehicles in subsection 1.1.1 and its current developments in our everyday cars in subsection 1.1.2.

1.1.1 Evolution of the automated vehicle towards the self-driving car

The research on the automation of the vehicle is an almost centenarian topic. Different phases of development and actors have succeeded each other. This part reminds, in a non-exhaustive way, of the main research and consortium projects led between academia and industrials along the years.

1.1.1.1 Phase 1: Research projects (1920s-1980)

One of the first contributions to driverless cars dates back to the 1920s at the Houdina Radio Control Company. They successfully proceeded with the demonstration of a car controlled by radio [Green 1925], see Figure 1.1a. A trailing vehicle sent radio signals to the front driverless vehicle in order to control its throttle, brake, and steering. In the 50s, General Motors in collaboration with RCA Labs demonstrated a vehicle system that was guided and controlled by wires that were laid in the pavement of a private road, promoting advances in highway design and transportation [PC6 1960]. In 1977, the Tsukuba Mechanical Engineering Laboratory in Japan developed an automated car, which identifies and avoids obstacles using two front cameras, at a velocity up to 30 km/h on private tracks [Tsugawa *et al.* 1979].

1.1.1.2 Phase 2: Consortium projects (1980-2000)

In the late 80s and 90s, research institutes and automotive manufacturers partnership numerous autonomous driving projects financed in. The European EUREKA Prometheus program [EUREKA], from 1987 to 1995, aimed to improve road traffic on both vehicles and infrastructure (e.g. driver assistance, Vehicle-to-Vehicle, V2V, and -Infrastructure, V2I, communication, driver monitoring). One of the major demonstrations was the twin cars VaMP and VITA II [Ulmer 1994] which drove over 1000 km on the French 3-lane A1-highway in a heavy traffic. They performed various driving tasks such as lane keeping, adaptive speed or distance keeping, lane changing, overtaking, and collision avoidance at a maximum speed up to 130 km/h using video cameras. However, this autonomous vehicle application was limited to well-structured environment and good weather/light conditions. At the same time,

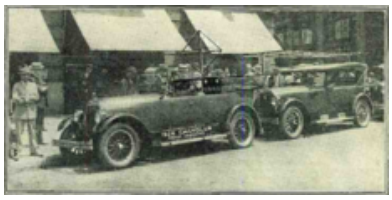
Carnegie Mellon University's Navlab succeeded in their project "No Hands Across America" with Navlab 5 [NHA]. They drove, from Pittsburgh, PA to San Diego, CA, USA, a steering controlled vehicle, with human-controlled throttle and brake. To quote but a few, others influential demonstrated projects were the ARGO project of University of Parma's VisLab [Broggi *et al.* 1999], a lane marking- and vehicle detection-based vehicle, which traveled the *MilleMiglia in Automatico tour* – a 2000 km journey along Italian highways –, and the platooning demonstration of automated highway in 1997 at San Diego by the California Program On Advanced Technology For The Highway (PATH) in collaboration with General Motors during the National Automated Highway System Consortium (NAHSC) [Rajamani *et al.* 2000], see Figure 1.1b.

1.1.1.3 Phase 3: Expansion of autonomous projects (2000-present)

Since the Defense Advanced Research Projects Agency (DARPA) organized autonomous vehicle competitions in 2004, 2005, and 2007, and thanks to new technologies, automated functions are evolving quickly and treat more complex scenarios in real environments. For the DARPA Urban Challenge (DUC) 2007 [DARPA], six teams succeeded in navigating through a city environment in traffic with both automated and human-driven vehicles. However, the vehicles were frequently manually paused for safety reasons, and a set of rules defined the environment and the expected situations that the vehicles had to face. The University of Parma's VisLab ran the VisLab Intercontinental Autonomous Challenge (VIAC) in 2010, a 15 900 km trip in 100 days from Parma, Italy to Shanghai, China [Broggi *et al.* 2013a], see Figure 1.1c. The VIAC objectives were to test the perception technologies and the vehicle's control system with a human-driven leader/automated follower approach. They extended their research by running the PROUD project [Broggi *et al.* 2014] in 2013 on open public urban roads and freeways through Parma, Italy, but with a specific path to follow, which was previously tested to solve the encountered challenges. In 2011, the European project HAVEit [HAVEit] demonstrated V2V and V2I communication up to 20 m and a joint system driver/co-system with automated assistance in road works and congestions, as well as temporary autopilot with driver monitoring. Meanwhile, CityMobil [CityMobil] European project tested cybercars for automated city transit, and the French project ABV [Resende *et al.* 2013] studied the sharing of driving between the driver and the assistance system at low speed. In 2012, the achievement of the SARTRE Project [SARTRE] showed highway driving in platooning with a three-car road train conducted by a human-driven vehicle on a proving ground. In 2013, another demonstration by Daimler and Karlsruhe Institute of Technology completed the Bertha Benz Memorial Route [Ziegler *et al.* 2014] with both city and highway driving in fully autonomous mode with a defensive driving style. The route was divided into intervals to ensure that operating times never exceeded 45 minutes. In 2016, the international Grand Cooperative Driving Challenge (GCDC) [Englund *et al.* 2016] presented three traffic scenarios for highways: merging two rows of vehicles into one, automated crossing and turning at an intersection, and automatically giving way to an emergency vehicle, using self-driving technologies, V2V and V2I communication.

Although, these projects have successfully demonstrated the implementation of an automated vehicle, their overall behavior is far inferior to the one of a human driver in the field of perception, prediction, and interpretation. Moreover, the requirements of the implemented

sensors need to be reduced in order to provide commercial roll-out.



(a) The radio-operated automobile from the Houdina Radio Control Co.

(b) Demo'97: I-15 lanes in San Diego, CA.

(c) The VIAC VisLab vehicle.

Figure 1.1: Illustration of the evolution of self-driving cars.

1.1.2 From ADAS to fully autonomous vehicles

The Advanced Driving Assistance Systems (ADAS) are active safety systems to inform or assist the driver in order to reduce and facilitate the driving tasks. We make the distinction between the automated vehicle, which presents automated functions with little or no direct human control, and the autonomous vehicles, which proposes a fully automated driving. Within this manuscript, the terms autonomous vehicle, autonomous car, self-driving car are equivalent. This part reports the framework of vehicle's automation, the use of ADAS in highway driving, and the evolution of the car-makers.

1.1.2.1 Levels of driving automation for on-road vehicles and adaptation of legislations

After Nevada, USA, authorized driverless vehicle experimentations on public roads in June 2011, other American states and countries (arranged in alphabetical order: Belgium, China, France, Germany, Italy, Japan, the Netherlands, Singapore, Spain, Switzerland or United Kingdom) adopted laws to test autonomous vehicles in traffic. In response to these evolutions, the United Nations Economic Commission for Europe (UNECE) amended the Vienna Convention on Road Traffic [United Nations] to make the use of automation more flexible. It allowed automated driving systems, "provided that these technologies are in conformity with the United Nations vehicle regulations or can be overridden or switched off by the driver" [UNECE].

With such a fast development in the automotive field, the Society of Automotive Engineers (SAE) published in 2014 a standard classification for automated and autonomous vehicles with a 6-level system, detailed in Table 1.1, from 0 (no control but active safety systems) to 5 (no human intervention for driving) [SAE]. Within this decomposition, the SAE provides a common taxonomy and definition for automation levels according to the operator, i.e. human driver or system, for the operational driving task (steering, acceleration, deceleration), the environment monitoring, the fallback performance, and the driving scenario capability. With respect to this taxonomy, the automated vehicle applies the levels 1 to 3 and the autonomous vehicle designates the levels 4 and 5.

Table 1.1: Automation levels from SAE International J3016 [SAE].

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

At the moment, main automakers propose level-2 functions for individual vehicles – except for Audi who implemented a level-3 feature for the traffic jam driving mode, and a level-4 for autonomous shuttle services, as it is described in the next section.

1.1.2.2 ADAS implemented in today vehicles for highway driving

The development of ADAS has three main objectives: (1) to increase safety inside and around the vehicle (crash avoidance and injury prevention), (2) to reduce its fuel consumption and ecological footprint, and (3) to provide driving comfort to the driver. In the scope of this work, only ADAS for the development of automated vehicles on highways are described, which provide a partial task of automation in the vehicle as aiding, warning or assisting features.

Nowadays, many technological evolutions in partial autonomous vehicles are already used in ADAS, and automakers now try to personalize ADAS to the driver's style [Hasenjäger & Wersing 2017]. Concerning highway planning, a driver can be assisted to maintain a constant speed with the Cruise Control function (CC) or not to exceed the speed limits with an Intelligent Speed Adaptation (ISA). Adaptive Cruise Control systems (ACC) adjust the vehicle speed in order to maintain a fixed distance, or a fixed time, to the front ego lane vehicle. In the scope of the development of automated and connected vehicles, the Cooperative Adaptive Cruise Control (CACC) would regulate the vehicle speed to the front ego lane vehicle considering the information on the acceleration of the next front ego lane vehicle, so that the system

speed is more stable and limits traffic jamming and fuel consumption; see [Wang *et al.* 2018c] for a recent review. While interacting with obstacles, a collision avoidance system is applied to either warn the driver of an imminent collision, or to autonomously take actions, such as braking, helping the driver to stay safe. The Forward Collision Warning (FCW) and Reverse Collision Warning (RCW) are visual and acoustic warnings for, respectively, front and rear safety distances with obstacles. Lane Change Assist (LCA) systems assess the surrounding traffic and warn the driver against impending collision during his/her lane change intent. Exploiting also the well painted lane marks on highways, systems based on Lane Departure Warning (LDW) alert if the vehicle is leaving the ego lane, or those on Lane Keeping Assist (LKA) keep the vehicle within the ego lane marks.

Nevertheless, all those functions are mainly reactive, as perception/action scenarios, and are used as isolated functions of the system. A systemic point of view including both predictive and reactive reasoning is still missing. That is part of the challenges for the next generation of self-driving cars. Moreover, two evolutions are conceivable: incrementally improving ADAS to turn them into self-driving cars, or thinking the self-driving car as a new system. The first option allows a progressive integration of the automation levels to the full automation without steering wheel. The second option considers the vehicle as a new fully autonomous system, where ADAS would only concern the dynamic control of the vehicle, warning systems and comfort of passengers. This remains an open question today among the car-makers and the new actors of IT as detailed in the next subsection.

1.1.2.3 The automotive companies and the emergence of new IT business

In addition to the research projects, many automotive companies unveil their semi-autonomous vehicles corresponding to levels 1 or 2 of the SAE classification (arranged in alphabetical order, non-exhaustive): Audi, BMW, Bosch, Continental, Daimler, Ford, General Motors, Hyundai, Mercedes-Benz, Nissan, PSA, Renault, Tesla, Toyota, Valeo, Volkswagen, Volvo. Level-2 automated functions, such as lane keeping, adjusting velocities to the front car and the road speed limits, or parking, are now available on commercial vehicles (e.g. Ford Fusion, Nissan QASHQAI, Peugeot 508, Tesla Model S). For the moment, only Audi proposes a level-3 Traffic Jam Assistant for its A8 L, see Figure 1.2a.

Nonetheless, one of the experiences with the highest cumulative traveled distance is the Google Car project, launched in 2011, which has driven over $16,000,000$ km at the end of 2018, see Figure 1.2b. Meanwhile, the transportation network companies Uber and Lyft collaborated with automotive companies to test their algorithms in order to offer a robo-taxi service. NuTonomy and Uber tested their taxi driver respectively in Singapore and Pittsburgh (PA)/Tempe(AZ), in 2016 and 2017, as well as Lyft, which launched an open platform in 2017 to give access to its ride-sharing network for real-world ride-sharing driverless cars scenarios.

Furthermore, the deployment of autonomous shuttle is increasing with Navya and EasyMile, which propose a public transportation for short distances in airports, universities or pedestrian and urban areas, see Figure 1.2c.

However, recent (deadly) accidents prove that the research and development of fully autonomous vehicles still requires several years of improvement, including robustness against their environment and driver relaxation, as well as a deeper reflexion on their social integration.

(a) Audi A8 L. (b) Waymo self-driving cars. (c) Navya autonom shuttle.

Figure 1.2: Illustration of automated and autonomous vehicles.

1.2 Motivation

The autonomous vehicle is one of the significant technological evolutions arousing popular enthusiasm in the past decade. As such, autonomous vehicles are now applied to different kinds of mobility, such as individual vehicles, on-demand robo-taxis, or autonomous shuttles used for the last kilometer or on limited areas. The development of this technology brings up potential improvements. First, it tackles the safety issue with systems which have faster reaction time than human drivers and are potentially able to communicate their intentions more clearly on the road than human drivers can do to each other. These system capacities would also help to decrease road congestion. In addition, they come along with a new vision of transport and relationship to the personal vehicle. This is part of a democratization and an environmental approach in order to reduce the number of vehicles in traffic, particularly in big cities, and to optimize the existing fleet by sharing it between several individuals (on-demand autonomous vehicles for car-sharing providing robo-taxi features). This would contribute to save transportation time and money. Furthermore, the social aspect is part of the motivation for vehicle automation by offering mobility to people with disabilities or difficulties to drive.

Among the various components of an autonomous system, the planning part is considered as the thinking part, responsible for the chosen actions. This part is in charge of making the link between environmental observations from perception and vehicle's actions, see Figure 1.3. In spite of the current trendiness of this topic, autonomous projects are not recent in the robotics literature and many reviewing works exist on motion planning strategies [Latombe 1991, Laumond *et al.* 1998, LaValle 2006] or, more recently, on the specific use case of autonomous driving [Katrakazas *et al.* 2015, González *et al.* 2016, Paden *et al.* 2016]. However, the standardization of the research to fit industrially realistic prototypes of self-driving cars introduces new concerns such as functional safety, real-time computing, a systemic approach, or low-cost developments, and specific popular applications for parking, intersection management, or highway driving. The motivation for highway driving lies in its simple driving structure and the driver's limited behavior in nominal situations, making it the most reachable context for the first fully autonomous systems in traffic. Furthermore, highways seem to be the first environment where drivers would be confident driving in a fully autonomous mode [Payre *et al.* 2014]. Nonetheless, the high velocity on highway environment is a major risk factor in case of crash.

The motivation of this Ph.D. work is to propose an approach to systemically treat decision making and trajectory generation in a dynamic highway environment. In addition, the state

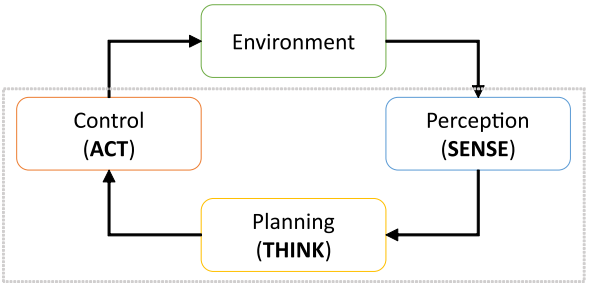


Figure 1.3: Illustration of the perception-planning-control autonomous system.

of the art lacks a taxonomy based on solutions that can be implemented with the available resources, but that are flexible enough to be able to evolve with future technological advances. Therefore, we assume that the vehicle's decision making is carried out in an "isolated" vehicle, without communication with its environment and realistic sensors' capacities for perception, which leads to a vision at $50 - 100$ meters. Even if these perception distances are admissible for highway behavior, their limitations are a major disadvantage for autonomous vehicles at high velocities. Moreover, we do not consider the driver in the loop at the core of the decision making and trajectory generation developments. Thereby, our work belongs to SAE level-4, with focus on highway driving. Nonetheless, the techniques used for decision making and trajectory generation are not dependent on the driver actions, and can be integrated without any intrinsic change in the SAE levels 1 to 3, where the driver will then affect the inputs, outputs and integration constraints. Consequently, the research question driving this work is to address the problem of a linear architecture between decision making and trajectory generation considering the uncertainties of perception, while avoiding an overly conservative vehicle's behavior.

1.3 Contributions

The problem addressed in this thesis is the motion planning, including the decision making and motion generation. The main contributions in the present dissertation are synthesized below.

1.3.1 A taxonomy of motion planning algorithms

The first contribution of this thesis is a state-of-the-art of the theoretical and experimental techniques of motion planning over the last decade for autonomous vehicles used as future every-day car. The research studies are multitudinous and varied. However, with respect to the recently published survey papers, the bibliographies were either too specific or too coarse, in the sense that a systemic approach for motion planning was missing.

The interest of this work is a survey of recent motion planning algorithms used for highway planning. A novel classification is also suggested, centered on intrinsic attributes of the methods oriented towards practical applications, enhanced with a radar charts representation. Furthermore, a synthesized comparison table is drawn which includes a comparison of methods, limited to the decision making, motion generation and deformation functions of

self-driving cars on highway environment. It follows two purposes: (1) motion planning techniques are highly dependent on the constraints of perception, control, and environment, thus it is important to refer to an algorithm with the appropriate behavior with relation to the target application; (2) a systemic treatment of motion planning will lead to develop complementary meta-algorithms. Finally, the issues of implementing motion planning algorithms on autonomous vehicle in mixed-human traffic infrastructure are overviewed for future research directions.

1.3.2 A new architecture for decision making and motion generation

Unlike previous planning schemes, which separate the motion generation and the decision making, our contribution addresses both the motion generation –under the form of trajectory generation– and the decision making –as an evaluation function– in a combined manner, in order to ensure the consistency between the choice of a maneuver and the calculation of the trajectory. The architecture proposed in this thesis provides an optimization of the motion generation using the decision making function as the evaluation function. In addition, real tests have shown that this architecture can be implemented in real time.

1.3.3 A continuous spatiotemporal maneuver generation

Our contribution is to optimize a trajectory generation process into a continuous space-time. Thus, we avoid the discretization of the solution by optimizing the evolution space for every motion of the ego vehicle. The decoupled optimization of position in a static environment and velocity in a dynamic one is also overcome as the evolution space is characterized in 4 dimensions (longitudinal position, lateral position, longitudinal velocity and time). Moreover, the proposed solution is set over a prediction horizon, i.e., the trajectory is predicted far ahead (5 to 10 seconds).

Our method defines non-collision intervals in the evolution space based on the adjacent obstacles. The intervals are characterized in space and time and then gathered into maneuvers. A simulated annealing algorithm is implemented to optimize, using the decision-maker as an evaluation function, the parameters of a sigmoid trajectory for each maneuver in order to find the near-optimal solution in a finite time. The flexibility and adaptability of this representation to the environment of the autonomous vehicle on highways are validated in real experiments.

1.3.4 An adaptive framework design for decision making

The objectives of the decision-maker are to select the future motion as the best compromise considering uncertainties and heterogeneous criteria of decision, such a acceleration, velocity, inter-distances, relative positions. Moreover, a deterministic decision allows to obtain replicable and justifiable results, which is necessary for a human critical safety process as autonomous driving. Our contribution is a high-level decision-maker providing a risk assessment by optimizing heterogeneous criteria, evaluating with arbitrary functions, and dealing with uncertainties.

The framework is made of three levels. The first one interprets the heterogeneous criteria to gather them in homogeneous categories using the fuzzy logic approach. The fuzzy logic

deals with vague or imprecise information, handles non-independent variables, and recalls human evaluation process. The second level combines the categories with the Dempster-Shafer Theory. It yields both the evaluation of decisions and how it has been obtained, fixes the influence of categories on the final decision, and interprets the fuzzy information into bound estimates. Finally, a linear combination of the beliefs intervals selects the best motion to follow.

Thus, the proposed framework for decision making simultaneously considers several factors of risk with their relative importance for human safety. It also handles personalization of the vehicle's behavior, depending on the passengers' risk perception and an aggressive or conservative driving style. Real testing validates the algorithm, which is evaluated by comfort and safety analyses.

1.3.5 Publications

Peer-Reviewed Journals

Title: A Review of Motion Planning for Highway Autonomous Driving
Authors: Laurène Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser
Journal: Transactions on Intelligent Transportation Systems, IEEE
Volume: early access, **Pages:** 1-23, **Year:** 2019. *Used in Chapter 2*

Title: Perception, information processing and modeling: Critical stages for autonomous driving applications
Authors: Dominique Gruyer, Valentin Magnier, Karima Hamdi, Laurène Claussmann, Olivier Orfila, and Andry Rakotonirainy
Journal: Annual Reviews in Control
Volume: 44, **Pages:** 323-341, **Year:** 2017.

Peer-Reviewed Conference Articles

Title: A study on AI-based approaches for high-level decision making in highway autonomous driving
Authors: Laurène Claussmann, Marc Revilloud, Sébastien Glaser, and Dominique Gruyer
Conference: IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Place: Banff, AB, Canada **Date:** October 5–8, 2017. *Used in Chapters 2 and 6*

Title: Multi-Criteria Decision Making for Autonomous Vehicles using Fuzzy Dempster-Shafer Reasoning
Authors: Laurène Claussmann, Marie O'Brien, Sébastien Glaser, Homayoun Najjaran, and Dominique Gruyer
Conference: IEEE Intelligent Vehicles Symposium (IV)
Place: Changshu, Suzhou, China **Date:** June 26-30, 2018. *Used in Chapter 3*

Title: Simulated Annealing-optimized Trajectory Planning within Non-Collision Nominal Intervals for Highway Autonomous Driving
Authors: Laurène Claussmann, Marc Revilloud, and Sébastien Glaser

Conference: IEEE International Conference on Robotics and Automation (ICRA)
Place: Montreal, QC, Canada **Date:** May 20-24, 2019. *Used in Chapter 4*

1.4 Structure of the document

The present Ph.D. thesis is organized in six chapters. The remaining chapters are described as follows:

Chapter 2, State of the Art This chapter presents a literature review of the motion planning algorithms applied to autonomous vehicles. Motion planning is considered under the decision making and motion generation functions. The scope of this review focuses on the highway environment. A taxonomy is proposed in order to highlight the main features of each algorithm for their application to real-time autonomous systems in complex environments. Thus, six families are introduced, considering their goal, space-time property and their mathematical domains. For each family, we review the basic idea, the specificities, the advantages and drawbacks, the improvements, and the interest for highway planning. At the end, a comparison table is built to summarize the aforementioned characteristics, intrinsic and extrinsic limits for a real-time use in highway motion planning.

Chapter 3, Decision making This chapter introduces our work for decision making. The decision part in motion planning is crucial and deals with many different kinds of criteria and data, such as guaranteeing the safety time-space between the ego vehicle and its environment, respecting the physical limits of the ego vehicle, taking into account the passengers' comfort, following the navigation orders, etc. This presents two main issues: the heterogeneous criteria and the uncertain data. The proposed framework for decision making addresses both by combining the fuzzy logic approach and the Belief theory.

Chapter 4, Trajectory generation A combined architecture for both trajectory generation and decision making is first introduced in this chapter. The main concern is to avoid two major problems of motion generation: the discretization of a set of solutions and the decoupled optimization of the static and dynamic environments. To do so, an optimization process uses the decision making as the evaluation function. Besides, the optimized variables are considered in a continuous spatiotemporal interval, which corresponds to the evolution space without collision. Three variables are optimized to generate a sigmoid trajectory, which fit properly the two main maneuvers on a highway environment, namely the lane changing and the lane following. Without any a priori assumption on the decision function, a metaheuristics simulated annealing algorithm is applied for the optimization process to deal with black box function. Secondly, non-collision nominal intervals are modeled as the search space for candidate solutions. They are defined as the free space between the obstacles on the adjacent lanes and characterized by collision-free velocity profiles and collision-free space maneuvers.

Chapter 5, Experimentation and results The experimental results are presented in this chapter. The validation tests are decomposed into six scenarios to model the different use cases encountered on highways depicted in Chapter 2. The scenarios are performed on

the prototype vehicle of Institut VEDECOM on a high speed test track with a 2-lane road, approximatively straight over 1000 meters. The planning algorithms have been integrated with a controller and perception modules. Two real obstacles (manual-driving) are introduced in the last scenarios.

Chapter 6, Conclusion, future work and research perspective This final chapter presents the conclusions on the main contributions of this Ph.D. work, as well as some possible future work. Also, perspectives on motion planning research are discussed.

Chapter 2

State of the Art

Self-driving vehicles raises relevant controversies, especially with recent deadly accidents. Nevertheless, they are still popular and attractive thanks to the improvement they represent to people's way of life (safer and quicker transit, more accessible, comfortable, convenient, efficient, and environment friendly). This chapter presents a review of motion planning techniques over the last decade with focus on highway planning. In the context of this thesis, motion planning denotes path generation and decision making. Highway situations limit the problem to high speed and small curvature roads, with specific driver rules, under a constrained environment framework. Lane change, obstacle avoidance, car following, and merging are the situations addressed in this work. After a brief introduction to the context of autonomous ground vehicles, the detailed conditions for motion planning are described. The main algorithms in motion planning, their features, and their applications to highway driving are reviewed.

Contents

The purpose of this work is to build a taxonomy of motion planning algorithms for highway driving for autonomous vehicles used like the workaday car of the future.

This chapter is organized as follows: Section 2.1 explains the scope of our classification, related to the perception and control constraints, and driver/highway driving rules, to highlight the specific situation of highway driving. Section 2.2 describes the state of the art, and

section 2.3 gives a comparison table for highway applications. Finally, section 2.4 introduces the present work with respect to the existing state of the art.

2.1 Considerations for highway motion planning

This section details the terminology related to the motion planning in subsection 2.1.1 and the motion planning scheme in subsection 2.1.2. The specificities of highway driving and the constraints on autonomous driving are listed, respectively, in subsections 2.1.3 and 2.1.4.

2.1.1 Terminology

Before dealing with planning algorithms, one needs to define the wide terminology involved. The adjective **ego** relates to the vehicle that is mastered and sensors-equipped. In contrast, other vehicles are denoted as **obstacles**. The kinematics of the vehicle are represented by its **states**, i.e. its position and orientation, and their time derivatives (position, velocity, and acceleration, linear and angular). The geometric state space is called the **configuration space**. The **evolution space** identifies the configuration space-time in which the ego vehicle can navigate. Both the configuration and evolution spaces are usually divided into three subspaces, see Figure 2.1: the **collision space**, in which the ego vehicle collides with obstacles; the **uncertain space**, in which there exists a probability for the ego vehicle to be in collision; and the **free space**, in which there is no collision. The generic term **motion** characterizes the states' evolution over time. Motion can refer in the literature to either **free-space** (spatial geometric zones), **path** (sequence of space-related states in the free space, i.e. geometric waypoints), **trajectory** (sequence of spatiotemporal states in the free space, i.e. time-varying waypoints), **maneuver** (predefined motion, considered as a subspace of paths or trajectories, i.e. motion primitives), or **action/task** (symbolic operations of maneuvers). We distinguish **generation**, which builds sequences of paths, trajectories, maneuvers, or actions, from **planning**, meaning the selection of one sequence among the generated motions. Finally, the **prediction horizon** denotes the space or/and time horizon limits for the simulation of the motion.

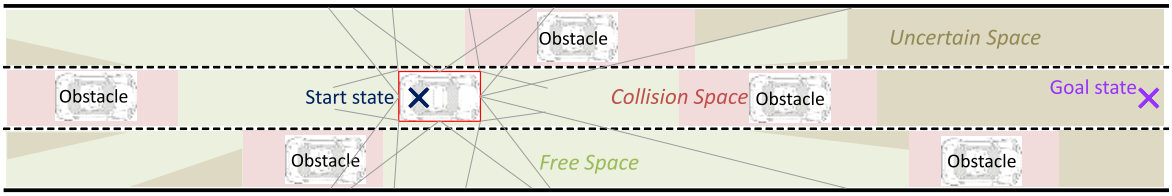


Figure 2.1: Evolution space representation: red = collision space, brown = uncertain space, green = free space. The evolution space decomposition is based on the perception data symbolized by the gray radii.

2.1.2 Motion planning scheme

A general abstraction of the hierarchical scheme for autonomous vehicles can be found in [Ziegler *et al.* 2014, González *et al.* 2016, Rodrigues *et al.* 2016]. We simplify and adapt the proposed scheme to the one depicted in Figure 2.2, which focuses on the motion strategy block.

The input data for the motion strategy block displays data for obstacles' behaviors and infrastructure description, obtained from the perception, localization, and communication (Vehicle-to-X - V2X) blocks. It does not pay any attention to how this information has been

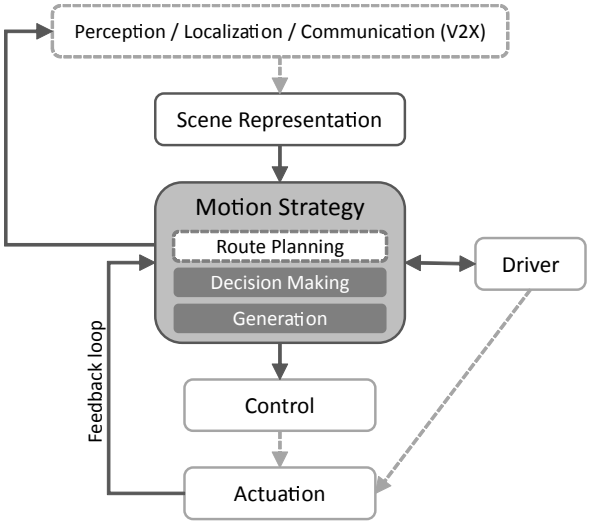


Figure 2.2: A hierarchical scheme of Autonomous Ground Vehicle systems.

collected, but to its quality such as the measures, their uncertainties, and their trustworthiness. Thus, a scene representation is introduced between the perception / localization / communication and motion strategy blocks, as presented in [Broggi *et al.* 2014]. This component manages sensors data and provides a perception map with obstacles, lanes, traffic, road, and ego information.

On the other hand, the control block is formally fed with the reference motion decided by motion strategy, and acts on actuators to move the ego vehicle. The motion strategy block also reacts as a closed-loop system on the control and actuation blocks. In fact, the information from the sensors on actuators' outputs provides an up-to-date ego state vector, so that the motion strategy remains accurate.

Another closed-loop system from the perception / localization / communication block to the motion strategy block is used to inform the ego perception and localization of the current and future motion. It also conveys the current and future ego motion intentions to the environment via communication.

Moreover, with semi-autonomous vehicles, the driver will interfere in the low levels of automation on the actuators pedals and steering, and on the high levels, so the motion strategy block acts more or less as a co-driver. In that sense, both communication from the driver to the motion strategy block and the reverse are necessary to warn the driver of the motion decision and to take into account his/her intention to drive. However, the driver will not affect the core of the algorithms in the motion strategy block. In fact, the interactions with the driver in the loop concern the input and output constraints (e.g. the driver wants to stay on the right lane), and the integration conditions (e.g. replanning if the driver modifies the planned trajectory). Thus, the presented motion planning methods can be extended to any SAE levels.

The motion strategy decides the most convenient motion according to some chosen criteria (addressed in subsection 2.1.4). In [Michon 1985], the author names the "three levels of skills and controls: strategical (planning), tactical (maneuvering), and operational (control) respectively." These three layers are largely exploited in the recent architecture [Broggi

et al. 2014, Ziegler *et al.* 2014]. The tactical part is usually divided into subparts, with at least one for behavioral planning and another for motion generation [Ziegler *et al.* 2014, González *et al.* 2016, Rodrigues *et al.* 2016, Gu *et al.* 2013]. According to the structures previously described, we distinguish five main functions of the motion strategy hierarchy, as shown in Figure 2.3: (i) route planning, (ii) prediction, (iii) decision making, (iv) generation, and (v) deformation. The space and time horizons for each of the motion planning functions are summarized in Table 2.1.

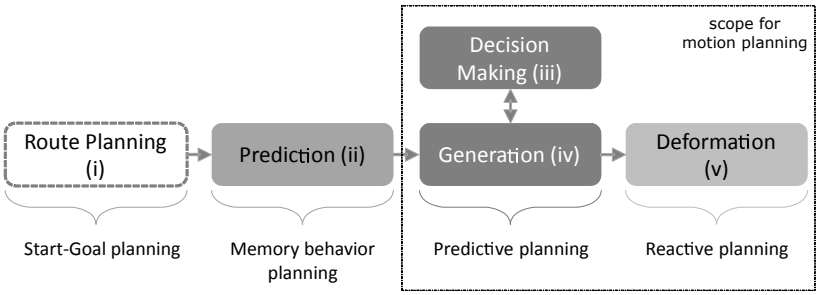


Figure 2.3: Motion planning functions. Motion planning acts as a global, local, and reactive motion strategy.

The route planner (i) is a trip scheduler; it provides a general long-term planning through the road network from the initial position to the driver's desired destination. This function is outside the scope of this work; see [Delling *et al.* 2009] for a related review.

The second function, the prediction step (ii), stores the current and historic dynamics data to predict the dynamics of all the elements surrounding the ego vehicle. This process allows to perform long-term risk estimation and dynamic replanning. The road infrastructure, the route direction, the driving rules, and the lane marking evolution are usually extrapolated from map information. The prediction of obstacles' behavior is the most critical task of the prediction function. Although it is necessary for motion planning, it is not explained in this work; see [Lefèvre *et al.* 2014] for a corresponding review.

The core functions addressed in this literature review are decision making (iii), generation (iv), and deformation (v), which we call the motion planning scope. In recent works [González *et al.* 2016], the motion planning approaches are organized as follows:

- A high-level predictive planning built around three objectives: criteria evaluation, risk minimization, and constraint submission (see subsection 2.1.4). Those are used for decision making (iii), i.e. to select the best solution out of the candidates' generation (iv). One either generates a set of motions and then makes a decision as a motion selector, or defines the behavior to adopt and then generates a motion to fit this behavior. This high-level stage benefits from a longer predicted motion but is time-consuming.

- A low-level reactive planning deforming the generated motion from the high-level planning according to a reactive approach, i.e. the deformation function (v). This acts on a shorter range of actions and thus has a faster computation time.

Table 2.1: Space and time horizon for the motion planning functions.

	Route Planning (i)	Prediction (ii)	Decision Making (iii)	Generation (iv)	Deformation (v)
Space	>100 m	>1 m <100 m	>10 m <100 m	>10 m <100 m	>0.5 m <10 m
Time	>1 min <1 hour	>1 s <1 min	>1 s <1 min	>1 s <1 min	>10 ms <1 s

2.1.3 Specificities of highway driving

Motion planning techniques highly depend on the use cases. Our considerations for highway driving are limited to lane-divided roads featuring unidirectional flow (opposing directions of travel being separated by a median strip) in nominal fluid traffic, i.e. with a dynamic speed over 60 km/h and without emergency maneuver. The road shape is made of straight lines, clothoids, and circles with small curvature. In a nominal situation, there are only motorized vehicles, which adhere to the same driving rules. Obstacle's behavior prediction is also limited to one-direction, two-lane changes – right or left – and to accelerate, maintain speed, or brake. Thus, traffic behaviors are more uniform than in city driving. We distinguish eight nominal non-exclusive situations during a highway trip, without focusing on exceptional situations:

- Lane keeping: The safety space in front of and behind the ego vehicle is guaranteed. The longitudinal decision is to maintain the desired speed, whereas the lateral decision keeps the ego trajectory inside the ego lane marks.

- Car following: Besides lane following, the ego vehicle must follow the front ego lane vehicle while maintaining its longitudinal safety distances.

- Lane changing: This decision is made under either directional or obstacles' constraints. The motion planner must ensure that the space in the target lane is sufficient and that the speed is adequate to keep the ego vehicle in a safe state.

- Lateral-most lane changing: To ensure fluid traffic flow and safety, some driving rules require leftmost or rightmost lane driving. Hence, the ego vehicle always seeks to change lanes until the lateral-most lane is reached.

- Passing: The ego vehicle respects a lane keeping or car following decision while obstacles are in the adjacent lanes. Keeping lateral security distances is required.

- Overtaking: This complex maneuver consists of a lane change, then passing an obstacle, and finally another lane change to return to the previous ego lane.

- Merging: Two actions occur on a highway: entering or exiting the highway, and merging two rows of vehicles into one. The ego vehicle must adapt its longitudinal and lateral speeds and distances to ease its way into traffic.

– Highway toll: The approach decision is to merge into a fictional lane in anticipation of a toll lane delimitation and to reduce the speed until stopping, whereas the leave decision is to accelerate to the reference speed and to merge into a real traffic lane.

The emergency maneuvers can either be treated from a motion planning point of view or from the control one. In the first case, two strategies are highlighted in the literature: applying reactive methods for a swerve maneuver to limit the collision, e.g. with a potential field approach in [Wang *et al.* 2019] (see subsection 2.2.2.3) or a graph search in [Jafari *et al.* 2017] (see subsection 2.2.2.2), and planning an emergency lane maneuver to drive the vehicle in a priori safe environments, as suggested in [Vanholme *et al.* 2010] with a predefined emergency lane trajectory. In the second case, the controller strategy aims to maintain the vehicle stability and minimize the collision states, as developed in [He *et al.* 2018].

Autonomous vehicles are also studied in platooning. This configuration decreases the distances between the vehicles and thus increases the capacity of roads. The motion planning strategy in platoons must be more robust than for individual vehicles in the sense of smoother acceleration and turns to guarantee the platoon's stability. Platooning is not as interesting from the perspective of motion planning as from that of control. We therefore do not consider any specific studies on platooning in this state of the art work. Please refer to [Axelsson 2017] for more details.

The main differences between highway, except for platooning, and city driving consist in a further look-ahead time, with a stronger focus towards the ahead direction of the road, whereas city driving involves a closer range of prediction but in all the directions. The highway dynamics is also simpler with lower turn-angle, no reverse, and less braking/acceleration, but higher and more constant velocity. Besides, vulnerable road users (e.g. pedestrians, motorcyclists) are less likely encountered on highways. Thus, even if there are less hazards, the risk due to high speed is stronger. Moreover, the higher distances imply poorer sensors capacities. Finally, less traffic insures more stable scenario. The algorithms which consider all these specificities in real-time will be favored for a practical application on highways.

2.1.4 Constraints on highway driving

Despite indicators such as high reliability, optimality, and low computation time needed for algorithmic specifications, we also consider more specific highway planning constraints. On the one hand, the environment's safety constraints respect the driving rules and avoid collisions. These are called hard constraints as they are absolutely essential for autonomous driving acceptance. On the other hand, the driver makes it necessary to respect ride optimization constraints for the minimization of time, distance, or energy consumption, and maximization of comfort. These are called soft constraints and can be relaxed. Other feasibility constraints rely on kinematic restrictions of the vehicle, which are the nonholonomic dynamics, i.e. the vehicle evolves in a three dimension space with only two degrees of freedom, a smooth path, i.e. the trajectory should be differentiable and its curvature continuous, and the dynamics limitations of a vehicle. The choice of the vehicle's model to handle these constraints induces algorithmic complexity: the more degrees of freedom are used, the more complex is the model solver. For most highway planning developments, there is no or very few (particle kinematics with longitudinal and lateral position and velocity states) consideration of the vehicle model, except for the explicit resolution of the potential field methods (see subsection

2.2.2.3) and the numerical optimization (see subsection 2.2.2.5). Indeed, the longitudinal and lateral constraints on the vehicle dynamics are important to maintain passenger's comfort for high speeds; thus simple dynamics models are sufficient. However, this question of degrees of freedom is a fundamental design parameter in motion planning and control architecture, and should be addressed to guarantee a safe and drivable motion [Schürmann *et al.* 2017], as well as consistency [Polack *et al.* 2018]. The last dynamic constraint is the evolution of the ego vehicle in time. To summarize, the authors in [Rodrigues *et al.* 2016] identify the quality requirements for the generated motion: "feasible, safe, optimal, usable, adaptive, efficient, progressive, and interactive."

2.2 Literature review of motion planning techniques applied on highway environment

The choice of the motion planning methods depends on the formulation of the motion planning problem.

Firstly, the problem formulation strongly varies with the inherited data (discrete/continuous, algebraic/analytic or static/dynamic) for the scene representation. Thus, the sensors' technologies are of great importance to implement a motion planning algorithm. Even if the uncertainties are more and more considered in the motion planning scope, too few papers addressing motion planning tackle the issue of the sensor's architecture and technology. We invite the readers to refer to [Gruyer *et al.* 2017] for a corresponding review.

Secondly, motion planning combines five unavoidable aspects, as seen in the previous sections: (i) state estimation, (ii) time evolution, (iii) action planning, (iv) criteria optimization, and (v) compliance with constraints. How these are handled changes the outlook of the problem.

Furthermore, two approaches for reviewing motion planning algorithms coexist: distinguishing and not distinguishing the driving modes. For the first, the driving modes and use cases are separated. For example, in [Bevly *et al.* 2016], the authors focus on the lane changing and merging maneuvers on highways. In this thesis, we consider a situation only as a specific set of criteria and constraints, not as a differentiating element for the algorithms. In this way, all methods could be applied to the different situations, with limitations to their proper functioning. Yet, we noticed that the main functions of motion strategy described in subsection 2.1.2 involve discriminating specificities among motion planning methods, e.g. the deformation function requires a reactive online real-time method.

In this section, we first explain our taxonomy in subsection 2.2.1, and then review the most popular approaches from their founding principles to their advancement for an application to highway planning since the DARPA Urban Challenge 2007 in subsection 2.2.2. We do not pretend to make an exhaustive taxonomy as an infinite number of ways of treating motion planning exist.

2.2.1 Taxonomy description

According to what we consider in section 2.1, we classify the algorithms in the families summarized in Figure 2.4. The term family refers to a set of algorithms that rely on the

same basic principle. They usually return the same type of output, qualify with the same attributes, and relate to the same mathematical domain. We propose a classification based on the following characteristics:

- The type of output: a space, a path, a trajectory, a maneuver, or a symbolic representation. This information should be adapted to the control and driver blocks' interface shown in Figure 2.2. In this respect, it is also essential to know whether the considered motion planning algorithm returns only a decomposition or a reference motion. In the first case, called a **set-algorithm**, a complementary algorithm should be added to find the feasible motion. The second case is identified as a **solve-algorithm**. These characterizations are extended from [Perez-Lozano 1983], where the author defines the *Findspace* and *Findpath* algorithms, respectively, to find a safe position in the evolution space, and to identify a sequence of safe positions that link the start and goal positions.

- The space-time property of the algorithm: the **predictive** or **reactive** horizon, as detailed in subsection 2.1.2.

- The mathematical domains: **geometric**, **heuristic**, **logic**, **cognitive**, **biomimetic**. This defines the philosophy of the approach and the theoretical framework of the solver. The **geometric** domain is based on the properties of space, and it directly works with the space constraints of the environment and ego vehicle (i.e. kinematic constraints). The subsequent problem is dealing with large space exploration and optimization. The **heuristic** domain depends on special knowledge, such as constraints or data correlation, about the problem. Usually, it is useful to solve more quickly, to find approximate solutions, to avoid algorithm complexity or an ego vehicle blockage situation. Yet, it is generally not sufficient for handling complex problems and does not guarantee that the optimal solution will be found. The **logic** domain refers to deductive approaches built on assertions. Such assertions are usually made on elementary rules driving the evolution of the environment. Their main advantage is that they easily link the effects to the causes, but they are subject to combinatorial explosion. **Cognitive** approaches rely on the evaluation of a situation based on prior knowledge on this specific situation and common sense, which is close to the logical way of thinking, for example adhering to driving rules. The main advantage of cognitive processes is their ability to use existing knowledge to gather new information. For the application to autonomous vehicles, the interest is in modeling the decision process formed on human behavior characteristics. It helps to justify the acceptability of autonomous vehicle behavior mimicking human behavior. However, we do not currently have enough experience to validate the effectiveness of such theories. Therefore, we do not consider algorithms that hinge on cognition as evidence. Many driver-based theories agree that driver behavior is too complex and too difficult, or even impossible, to model, as it is not rational. The last domain we would like to explore is the **biomimetic** domain, which describes physics-inspired approaches. They obey the physical laws, which are convenient to implement but can be stuck in infinite loop motion behaviors. The convergence of the system has to be obtained with feasible solutions.

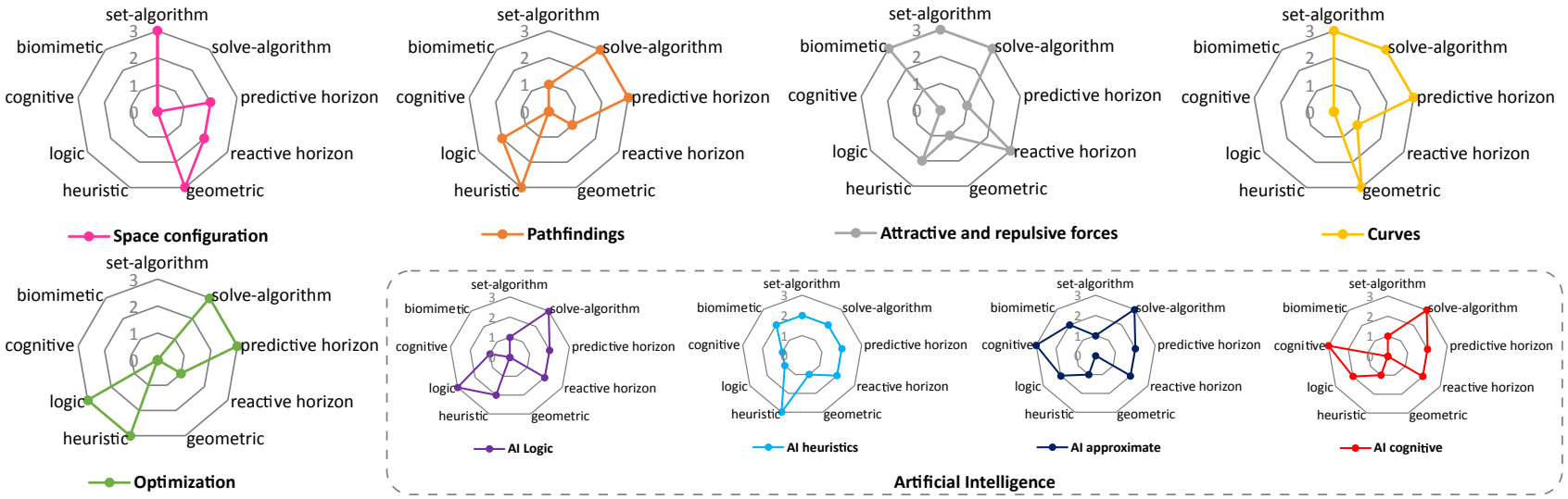


Figure 2.4: Taxonomy classification. The radar charts show the distribution of each of the families considering the selected attributes. At the center of the chart, the methods do not suit the corresponding attribute at all (value = 0). In contrast, an outer position of the attribute means it is well adapted to the use of the method (value = 3). Each value is assigned according first to the definitions of the families, and second to their occurrence in the literature.

2.2.2 Algorithm classification

For each family, we review the basic unifying idea, the specificities, the advantages/drawbacks, the evolution, and the interest for highway planning. One notices that the solutions were first dealing with a specific method in a specific scientific field, and then evolved to multi-field mixed algorithms.

2.2.2.1 Space configuration

The space configuration analysis is the choice of a decomposition of the evolution space. It is a set-algorithm used mostly for motion generation or deformation when specified. The methods are based on geometric aspects; they refer to either a predictive approach with a coarse decomposition to limit computational time, or a reactive approach with a finer distribution to be more accurate. The main difficulty is finding the right space configuration parameters to obtain a good representation of motion and environment [Kavraki *et al.* 1996]. If the discretization is too coarse, the collision risk will be badly interpreted and it will not be possible to respect kinematic constraints between two successive decompositions; however, if the discretization is too fine, the algorithm will have poor real-time performance. We distinguish three main subfamilies for space decomposition, illustrated in Figure 2.5: sampling points, connected cells, and lattice. The basic idea is:

1. to sample or discretize the evolution space;
2. to exclude the points, cells, or lattices in collision with obstacles or not feasible; and
3. to either send those space decompositions as free-space constraints, or to solve the resulting space configuration with a pathfinding algorithm (see subsection 2.2.2.2) or a curves planner (see subsection 2.2.2.4) to directly send waypoints, connected cells sets, or lattice sets to the control block.

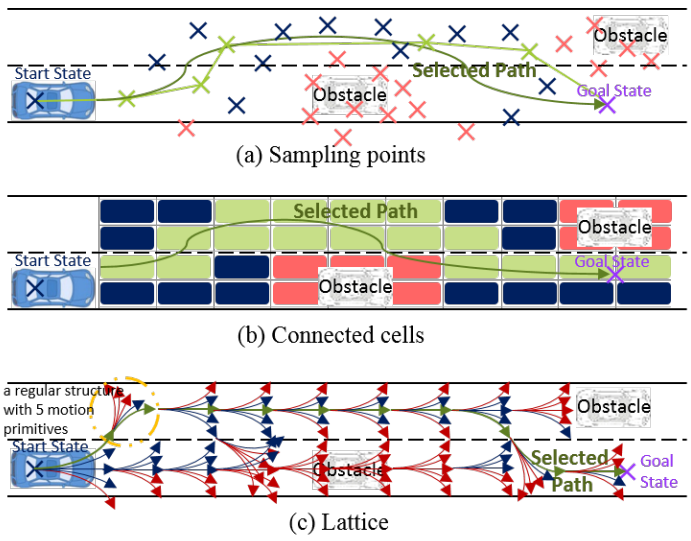


Figure 2.5: Illustrations of space configurations: (a) sampling points, (b) cells decomposition, (c) lattice. The points / cells / maneuvers are candidate, dismissed and chosen solutions in blue, red, and green.

Sampling-Based Decomposition (Figure 2.5a) Two main routines are used in the literature to return a set of sampling points. The first one chooses the points in the evolution space of the ego vehicle with respect to the kinematic constraints, but with higher calculation time due to the optimization choice of the samples under these constraints. The second one picks random points in the evolution space, so that the algorithm is computed faster but the method is incomplete, not replicable and sensitive to the random points' distribution. Moreover, the links between two points are not necessarily kinematically feasible. Both routines are suboptimal and do not guarantee that a solution will be found if one exists, or that a solution will be returned in a finite computational time. However, sampling configurations are flexible to dynamic replanning and do not require any explicit modeling of the collision space. They can thus be used for a reactive trajectory deformation.

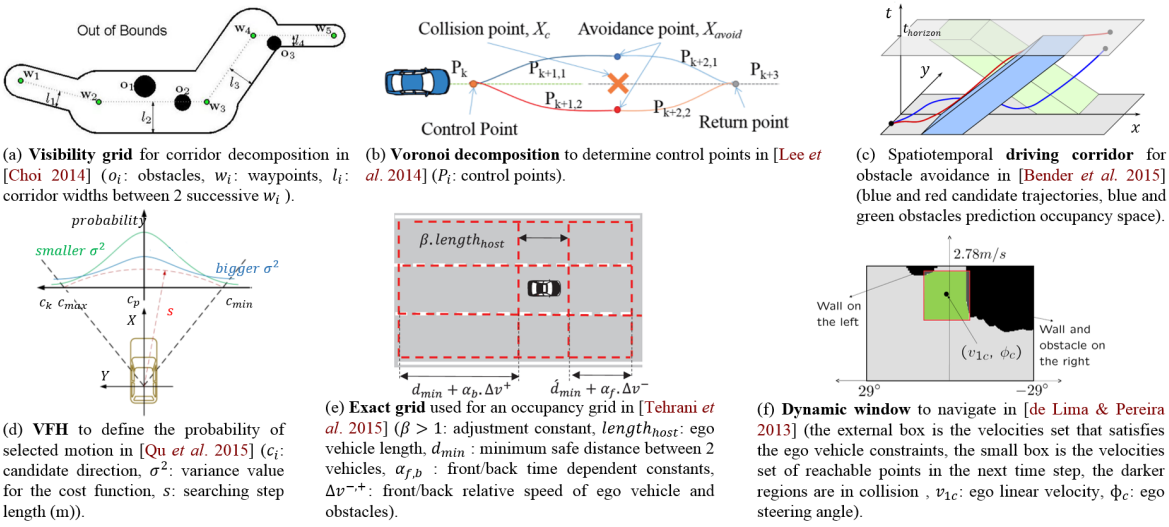
The most popular random method is the **Probabilistic RoadMap** (PRM) [Kavraki *et al.* 1996]. It uses random samplings picked in the evolution space during the construction phase. These samplings are connected to their neighbors to create an obstacle-free roadmap, which is then solved during a second query phase by a pathfinding algorithm, e.g. Dijkstra (see subsection 2.2.2.2) in [Villagra *et al.* 2012]. In [Li *et al.* 2014], the authors first sampled the configuration space based on a reference path, e.g. the ego lane centerline, selected the best set of sampling points according to an objective function, and then assigned a speed profile to the path to respect safety and comfort criteria.

A better strategy is to consider both space and time dimensions in the decomposition. Dealing with **spatiotemporal sampling** points makes it possible to obtain a predictive algorithm, as was done in [Hesse *et al.* 2010], where points are constrained in a 5-dimensional evolution space with vehicle position, orientation, velocity, and arrival time. Considering the drawback of tiny space and space close to obstacles, adaptive samplings can be adapted to autonomous vehicles, as presented in [Hsu & Sun 2004] for robots from 2 to 8 degrees of freedom.

These methods are usually preferred for mobile robotics or autonomous vehicles in unstructured environments. Their use for highly structured highway planning is thus diminished.

Connected Cells Decomposition (Figure 2.5b) The methods first decompose the space into cells using geometry, and then construct an occupancy grid and/or a cells connectivity graph, see Figure 2.6 for application examples. In the occupancy grid approach, a grid is generated around the ego vehicle. The information on the obstacles' detection is overlaid on the grid. In case uncertainties are considered, stochastic weights are added to the cells to obtain a cost-map representation. The main drawbacks of an occupancy grid are the large memory requirements and high computation time, the false indicative occupation with moving obstacles, and a space- and time-varying resolution. In the connectivity graph approach, the nodes represent the cells, and the edges model the adjacency relationship between cells. The graph can be interpreted as a path along the edges of the cells or a path to find inside the connected cells.

Two strategies are distinguished: those that are based on the obstacles, and those that are not. The representation of obstacles plays a key role for cells decomposition algorithms. Obstacles are usually represented as convex polygons [Kuan *et al.* 1985], rectangular [Liu *et al.* 2017a], triangular [Nilsson *et al.* 2017], circular [Song *et al.* 2013, Choi 2014, Mouhagir



(a) **Visibility grid** for corridor decomposition in [Choi 2014] (o_i: obstacles, w_i: waypoints, l_i: corridor widths between 2 successive w_i).

(b) **Voronoi decomposition** to determine control points in [Lee *et al.* 2014] (P_i: control points).

(c) Spatiotemporal **driving corridor** for obstacle avoidance in [Bender *et al.* 2015] (blue and red candidate trajectories, blue and green obstacles prediction occupancy space).

(d) **VFH** to define the probability of selected motion in [Qu *et al.* 2015] (c_i: candidate direction, σ^2: variance value for the cost function, s: searching step length (m)).

(e) **Exact grid** used for an occupancy grid in [Tehrani *et al.* 2015] ($\beta > 1$: adjustment constant, $length_{host}$: ego vehicle length, d_{min} : minimum safe distance between 2 vehicles, $\alpha_{f,b}$: front/back time dependent constants, $\Delta v^{-,+}$: front/back relative speed of ego vehicle and obstacles).

(f) **Dynamic window** to navigate in [de Lima & Pereira 2013] (the external box is the velocities set that satisfies the ego vehicle constraints, the small box is the velocities set of reachable points in the next time step, the darker regions are in collision , v_{1c}: ego linear velocity, ϕ_c: ego steering angle).

Figure 2.6: Illustrations of connected cells decompositions.

et al. 2016], ellipsoidal shapes [Claussmann *et al.* 2015, Wang & Ayalew 2016], based on the obstacle dimensions and speed; or the entire road lane [Yu *et al.* 2016]. For non-obstacle-based representation, the cells' organization can be determined offline, and then filled online. The grid is rapidly obtained, but does not take advantage of the environment properties. On the other hand, the obstacle-based representation builds an online grid, which is more computational as replanning is necessary to consider the dynamic of the environment. Whatever the decomposition, probabilistic occupancy can be applied to define a more realistic decomposition proportional to the probability of occupancy, as proposed in [Moras *et al.* 2011].

The most intuitive non-obstacle-based method is the **exact decomposition** (Figure 2.6e), which separates the space with vertical and/or horizontal segments. An egocentric exact decomposition adapted to the vehicle's size and the obstacle's relative speed is used to exploit an occupancy grid for a Markov Decision Process in [Mouhagir *et al.* 2016] and a lane change flow chart in [Tehrani *et al.* 2015] (see subsection 2.2.2.6). Reference [Yu *et al.* 2016] fills the occupancy grid with a belief state from the semantic lane information. In [Ardelt *et al.* 2010], 8 overlapping static and dynamic cells are mapped based on the lane configuration and performance of integrated sensors. Considering the nonholonomic behavior of the vehicle, the curvilinear or **polar grids** [Moras *et al.* 2011] provide a more realistic decomposition around the ego vehicle. However, as non-obstacle-based occupancy grids cannot evolve according to the obstacles' dynamics, they quickly become obsolete due to the computational cost of a refined decomposition. Another drawback is the lack of accuracy on obstacles' positions.

To consider environment uncertainties, the **Vector Field Histograms** (VFH) from [Borenstein & Koren 1991] statistically models the evolution space with a velocity histogram occupancy grid as a polar histogram (Figure 2.6d). Reference [Qu *et al.* 2015] proposes a constrained VFH to treat kinematic and dynamic constraints to extend to highway applications. VFH methods are mostly used for deformation function as a reactive planner, as they are robust to sensors' uncertainties [Lee *et al.* 2014].

To reduce the search space, decrease the computation time and deal with dynamic obstacles that are either hard to model or not modeled, the **Dynamic Window** (DW) approach,

introduced in [Fox *et al.* 1997], reduces the search space into a reachable velocity space within a short time interval (Figure 2.6f). It is also used as a reactive planner for trajectory deformation. In [Kang *et al.* 2015], DW is directly applied on an image from the camera. Reference [de Lima & Pereira 2013] combines a DW approach for avoiding unmodeled obstacles with a Velocity Vector Fields (VVF) (see subsection 2.2.2.3). The authors in [Mitsch *et al.* 2013] demonstrate safety recommendations for both static and moving obstacles using DW.

The routines of the second type use the obstacles to set the cells decomposition. The **Voronoi decomposition** [Kuan *et al.* 1984] builds cells between particular points, which represent each obstacle (Figure 2.6b), mainly using the Euclidean bisection (L2 Euclidean distance norm) or absolute value (L1 Manhattan distance norm). The method is extended to polygonal obstacles with the generalized Voronoi diagram [Aurenhammer 1991]. The decomposition generate a non-regular grid and is classically interpreted as the way that is equidistant from each obstacle (way along the edges), i.e. a skeleton. Reference [Villagra *et al.* 2012] uses Voronoi diagrams to reduce the obstacle-free point's space, whereas the authors in [Lee *et al.* 2014] propose a real-time algorithm based on one Voronoi cell built on the next collision point in the ego vehicle trajectory. To increase the distance between an obstacle and the ego vehicle, weighted [Sugihara 1993] and uncertain [Ok *et al.* 2013] Voronoi diagrams have been developed in mobile robotics, but they are not that common for autonomous vehicles. The major drawbacks of Voronoi decompositions are the heterogeneous cells sizes, the kinematic feasibility of linking adjacent cells, and the dynamic evolution, which involves time-consuming replanning. Moreover, the equidistance does not necessarily guarantee safety. In response to these issues, the **approximate methods** split up into thinner cells when the obstacles are closer to obtain a more accurate occupation grid. A classic approximate approach is the quad tree decomposition, as in [Trepagnier *et al.* 2011]. Building a dynamic cells decomposition remains the major drawback of the Voronoi and approximate approaches.

A major improvement is then to add a time dimension to the spatial decomposition. The **visibility decomposition** (Figure 2.6a) picks points of interest in the scene representation, and links them with segments if these do not intersect obstacles [Lozano-Pérez & Wesley 1979]. They are extended to a path velocity representation in [Kant & Zucker 1986]. The points of interest can be based on the obstacles' vertices [Johnson & Hauser 2013] or along the road border [Choi 2014]. This implies that the path along the edges can be in contact with obstacles, which is then adapted to the highway planning problem by space shifting the trajectories, as in [Johnson & Hauser 2013]. In addition, the **driving corridor** representation uses road boundaries and the spatiotemporal positions of obstacles to produce a set of free spatiotemporal evolutions as corridors [Ziegler *et al.* 2014]; see Figure 2.6c. From a static decomposition based on visibility decomposition as graph connectivity, [Choi 2014] builds a dynamic corridor from **Velocity Obstacles** (VO) analysis. This spatiotemporal algorithm returns the set of all the velocities of the ego vehicle that lead to a collision [Fiorini & Shiller 1998]. A velocity outside the spatiotemporal representation guarantees that there will be no collision under the hypothesis that the obstacle velocity prediction is correct. The passage and region channels are then solved with a connectivity graph [Kuan *et al.* 1985] or with a space constraint-based optimization method [Gu *et al.* 2016b]. Reference [Bender *et al.* 2015] builds the corridors with the homotopic method to enumerate the possible maneuver variants from a path velocity decomposition and [Altché & De La Fortelle 2017]

proposes a four collision-free cells partitioning to design a spatiotemporal transition graph. A major drawback of driving corridor algorithms is still their calculation time.

Most static cells decompositions are of little interest in the context of highway planning. Methods that are only based on obstacle decomposition exclude the road geometry, which is at the core of highway driving. In contrast, algorithms that benefit from both road constraints and dynamic obstacles are of major use for highway planning.

Lattice Representation (Figure 2.5c) In motion planning, a lattice is a regular spatial structure, which is a generalization of a grid [LaValle 2006]. It is possible to define motion primitives, that connect one state of the lattice exactly to another. All the feasible state evolutions resulting from the lattice are represented as a reachability graph of maneuvers. The main application of lattice methods is predictive planning. The advantage of a lattice representation is the consideration of the kinematic constraints implicitly handled by the motion on the lattice [Pivtoraiko *et al.* 2009], as well as a spatiotemporal consideration. Moreover, the lattice can be calculated offline for a quick replanning [Schubert *et al.* 2008]. Unfortunately, their application to reactive planning is mostly limited due to the fixed structure.

The classic lattice representation is based on the **maximum turn strategy** [Broggi *et al.* 2014, Schubert *et al.* 2008], where only the turning radius of the ego vehicle is discretized to propose different curved paths. As an improvement, the velocity (speed and acceleration) is considered with a **curvature velocity** method as presented in [Ko & Simmons 1998], and extended to autonomous vehicles in [Gu *et al.* 2016b, Constantin *et al.* 2014]. The main drawbacks of these methods are the rigidity of the predefined motion set and the high density of the motion graph required to reach the goal position. Nevertheless, it is possible to define an **environment-adapted lattice**, in contrast to the previously discussed lattices based on predefined motion. The authors in [Gu *et al.* 2013, Ziegler & Stiller 2009, McNaughton *et al.* 2011, Xu *et al.* 2012] operate regular sampling points over the spatiotemporal evolution space based on highway lane marks and centerlines. The use of curves to connect the sampling points provides a curved lattice graph set-algorithm. Other approaches adapt the lattice to the driver's behavior for a priori maneuver, as done in [Yao *et al.* 2012] for lane changing.

Lattice representations compile both road boundaries and kinematic constraints, and can be quickly replanned, which is useful for highway planning. Nonetheless, the structure's iteration memory requirement and long-term advantage represent a burden for fast computing path planning on a highway. As will be presented in subsection 2.2.2.4, curve iterations as tentacles are favored over lattice representation for highway motion planning.

2.2.2.2 Pathfinding algorithms

The pathfinding algorithm family is a subpart of graph theory in operational research used to solve combinatorial problems under a graph representation. The graph can be weighted or oriented with sampling points, cells, or maneuvers nodes. The basic principle is to find a path in a graph to optimize a cost function. Traveled distance, fuel consumption, and comfort are the main cost functions for highway planning problems [Villagra *et al.* 2012, Gu *et al.* 2016b, Bahram *et al.* 2014]. The graph resolution is based on logic and heuristic methods, which are mainly solve-algorithms and refer to the decision function even if they do not apply any decision but a selection. The subfamily of Rapidly-exploring Random

Trees (RRT – see below for details) comprises both set- and solve-algorithms, with motion generation and selection. The main use of pathfinding algorithms is for route planning, but they adapt well for local planning and applications to highways as predictive algorithms. As benefits, these algorithms are universal and widely used, and solve either known or unknown environments. The main drawbacks are their dependency on the graph size and complexity, which affects the choice of the solver, and their need for detailed information on the space configuration, which makes them slow in vast areas. We will restrict our review to the most frequently observed algorithms for highway autonomous driving.

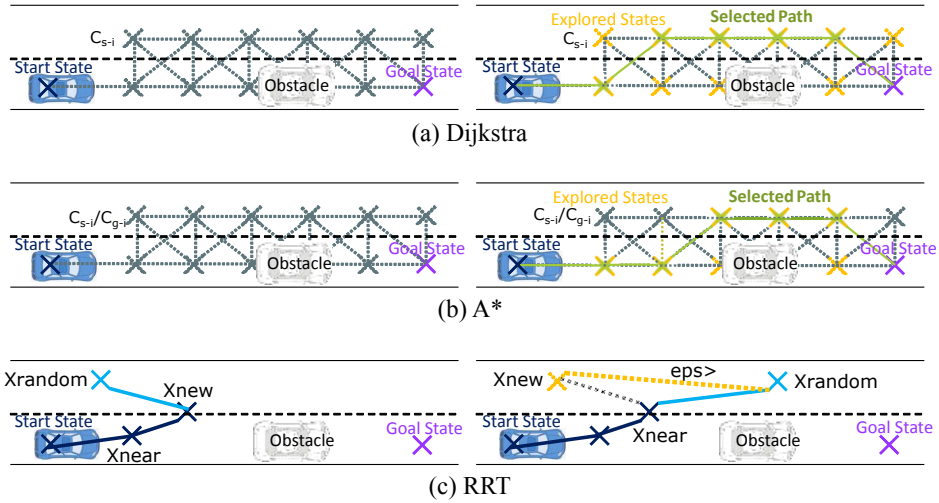


Figure 2.7: Illustrations of the processes of (a) Dijkstra, (b) A*, and (c) RRT.

For known environments, the graph is previously generated by a space configuration algorithm to model the connectivity of the evolution space (see subsection 2.2.2.1). In the **Dijkstra** algorithm from [Dijkstra 1959], the author details a method to "Find the path of minimum total length between two given nodes P and Q", which becomes a very popular graph solver for motion planning application to autonomous vehicles [Villagra *et al.* 2012, Gu *et al.* 2016b, Bahram *et al.* 2014]; see Figure 2.7a. As the algorithm uniformly explores all the directions, it finds the optimal path with respect to the cost function, but its computational time is high.

This drawback was first reduced with the **A*** algorithm introduced in [Hart *et al.* 1968], and recently tested on autonomous vehicles' replanning in [Bounini *et al.* 2017]; see Figure 2.7b. It consists of applying Dijkstra's algorithm with a heuristic search procedure on the goal-node to expand the fewest possible nodes while searching for the optimal path. The heuristic should always be optimistic, i.e. the real cost should be higher than the heuristic cost, as otherwise the minimal path will be distorted. As an example of a heuristic evaluation function, [Schubert *et al.* 2008] chooses the distance to both ego lanes' borders and the traveled distance for Dijkstra's cost function, whereas [Boroujeni *et al.* 2017] adds to the travel time function the distance to the goal and hazardous motions penalizations as heuristics. It is also possible to weight the heuristic function to reduce the calculation time, as described in the **Anytime Weighted A*** (AWA*), which guarantees the finding of a solution with a non-admissible heuristic [Hesse *et al.* 2010]. Besides, when considering kinematic constraints, the approach of **hybrid-state A*** search in [Dolgov *et al.* 2008] applies a first heuristic to

consider nonholonomic constraints, and then a second dual heuristic that uses an obstacle map.

A further disadvantage of Dijkstra and A* stems from dynamic environments. In fact, at each time step, the graph has to be reconstructed. To avoid a high time calculation and dealing with partially known environments with dynamically changing weights, a heuristic improvement consists of a dynamic cost graph search, as does the **D*** algorithm [Stentz 1994] in [Rezaei *et al.* 2003].

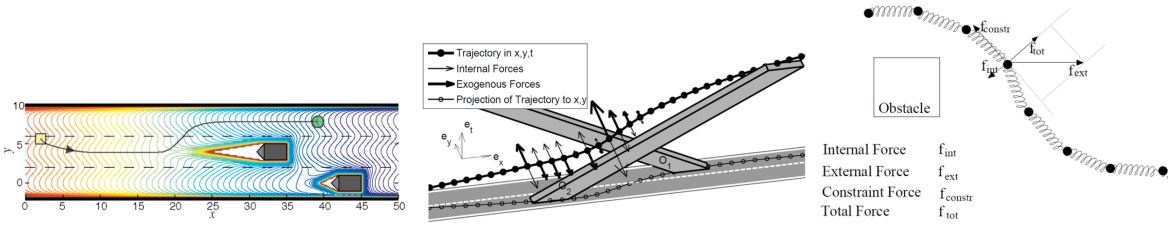
For unknown environments, the **RRT** subfamily [LaValle 1998] constructs its own nodes in the evolution space, as illustrated in Figure 2.7c. The reasoning is close to the PRM, except that the nodes are built from one to another and the output is a path (to solve the nodes connection if a non-collision and kinematically path exists). Thus, it guarantees kinematic feasibility and can be used for a reactive generation. Authors in [Ma *et al.* 2015] demonstrate a fast RRT for replanning trajectory. As for Dijkstra's algorithm, there are a large number of evolutions for RRT algorithms in mobile robotics [Connell & La 2017], but currently few applications for highway driving, such as the example in [Hwan Jeon *et al.* 2013], which looks at more efficient nearest-neighbor techniques with probabilistic optimality in **RRT***. For a randomized graph, the main drawback is the randomly collected sampling nodes, which may result in a poor connectivity graph and no replicability. A simple way to increase the connectivity is to add a probability function of generating intermediate points in a specific area, as done by the authors in [Villagra *et al.* 2012].

Similarly to sampling-based decomposition, probabilistic graph search is not well suited to a highway structured environment. Besides, the highway is usually a known environment, easily represented with space configuration algorithms in subsection 2.2.2.1. In that sense, deterministic pathfinding is favored in highway motion planning for autonomous vehicles.

2.2.2.3 Attractive and repulsive forces

The attractive and repulsive forces approach is a biomimetic-inspired method. The evolution space is symbolized as attractive forces for desired motions (e.g. legal speed), and repulsive forces for obstacles (e.g. road borders, lane markings, vehicles). The main advantage is thus to be reactive to the dynamic evolution of the scene representation. The motion of the ego vehicle is then guided by the resultant forces vector, so no explicit space decomposition is needed. Reference [Bounini *et al.* 2017] shows how parabolic and conical functions are well suited as potential functions. The resolution of the resultant vector is achieved by either a gradient descent method [Bounini *et al.* 2017, Galceran *et al.* 2015] – a simple resolution without a vehicle model, or by the application of Newton's second law [Gerdes & Rossetter 2001] – based on a vehicle model, which provides a feasible motion under kinematic constraints. The attractive and repulsive forces approach both sets and solves the motion planning problems in a continuous space representation. As the modeling of all the evolution space is time-consuming, these algorithms are mostly used as a reactive motion deformation.

The **Artificial Potential Field** (APF) concept [Khatib & Le Maitre 1978] was first introduced to real-time mobile robotics in [Khatib 1986]. Reference [Wolf & Burdick 2008] adapts a set of four artificial potentials over lanes, road, obstacles, and desired speed, to model the highway functions described in subsection 2.1.3; see Figure 2.8a. In [Reichardt & Shick 1994], the authors use a framework of electric fields as a risk-map with weighted



(a) APF in [Wolf & Burdick 2008]. The yellow square denotes the starting position and the green circle the end position. Lower potentials are in blue, higher in red.

(b) VFF in [Hesse *et al.* 2010]. O_1 and O_2 are moving obstacles.

(c) Elastic Band Model in [Gehrig & Stein 2007].

Figure 2.8: Illustrations of attractive and repulsive forces approaches.

partial potential to distinguish between emergency reactions and preventive actions. The benefit to time consideration with a velocity potential leads the ego vehicle to progress forward smoothly, as emphasized in [de Lima & Pereira 2013] with the use of a **Velocity Vector Field** (VVF). Furthermore, APF returns direct control inputs [Gerdes & Rossetter 2001] or constraints for the optimization solver [Liu *et al.* 2017a, Rasekhipour *et al.* 2017]. The first drawback for application to vehicle planning is the oscillatory behavior when close to obstacles, but smoothing algorithms overcomes this [Galceran *et al.* 2015]. The second difficulty is the presence of local minima. Not only is the ego vehicle stuck in a local minimum, but overcoming this issue impacts the smoothness of the path and calculation time. A trivial solution is to add a heuristic to exit the minima with a randomized path, as is done in [Bounini *et al.* 2017], where local minima are also repulsive artificial potentials. This tends towards the elimination of the local minima afterwards. However, local minima might be necessary in highway planning to keep the ego vehicle safe from inopportune lane changes, as stated in [Wolf & Burdick 2008].

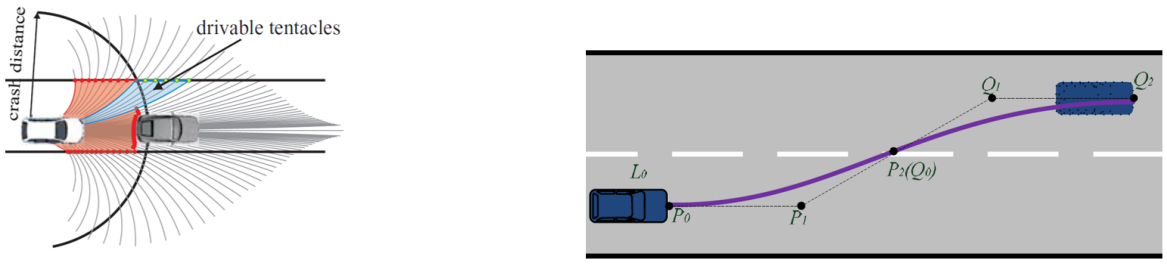
Extensions to uncertain environments allow to transpose the algorithms from a deterministic robotics environment to highway planning. The **Virtual Force Field** (VFF) introduced by [Borenstein & Koren 1991] uses the VFH decomposition (see subsection 2.2.2.1) and interprets the probability distribution as potential forces to guide the vehicle along the weakest grids. This method is adapted to highway driving in [Hesse *et al.* 2010] to deform a trajectory previously obtained with sampling points and A* methods; see Figure 2.8b.

Another drawback of the APF highlighted by [Khatib 1986] is the lack of dynamic reasoning: namely, only spatial dimensions are used for dynamic obstacle avoidance, so that the ego vehicle always tried to avoid the obstacle by bypassing it. The **elastic band** algorithm [Quinlan & Khatib 1993] models the environment as a spring-mass system, considering N discrete nodes on which potential forces are applied [Song *et al.* 2013]; see Figure 2.8c. Reference [Gehrig & Stein 2007] applies elastic band to car following as a deformation of the leader vehicle path and [Keller *et al.* 2014] develops a time-related elastic band framework with temporal waypoints for lane changing.

Attractive and repulsive forces are widely implemented for reactive planning in all kinds of robotics. Except the high time calculation, they prove their efficiency for highway planning, thanks to a straightforward integration of the scene representation, provided that the potential functions are well chosen within the environment.

2.2.2.4 Parametric and semi-parametric curves

Parametric and semi-parametric curves are major geometric methods in path planning algorithms on highway for at least two reasons: (i) the highway roads are built as a succession of simple and predefined curves (line, circle, and clothoid [Baass 1984]); and (ii) a predefined set of curves is easy to implement as candidate solution sets to test. Moreover, as some of the curve-based algorithms directly take into account the kinematic constraints of the vehicle, they are widely used to complement other methods. Geometric considerations are usually separated from the dynamic constraints. For instance, the authors in [Villagra *et al.* 2012, Lima *et al.* 2015b] first solve a static problem considering the geometry of the road, and then solve the dynamic dual problem by searching for a speed profile built to adapt to the curvature profile of the road and to respect the constraints of the dynamic obstacles. In contrast, [Vanholme *et al.* 2013] first fixes the speed profile and then deforms the curve path. Those decoupled approaches could lead to a long time reaction for the vehicle in case of a blocking situation or replanning. Otherwise, curve planners are well suited to a predictive approach too, as they define a motion from a start to a goal point or cell or maneuvers. This property is also suitable for replanning stages.



(a) Tentacle scheme in [Von Hundelshausen *et al.* 2008].

(b) Bézier curve in [Chen *et al.* 2013]. P_i, Q_i are control points.

Figure 2.9: Illustrations of the point-free and point-based curves methods.

We distinguish between two exploitations of curve algorithms. First, the point-free curves subfamily is used to build kinematically feasible trajectories as a set of candidate solutions (maneuvers). Second, the point-based subfamily uses curves to fit a set of chosen waypoints (sampling points or cells). The first routine is a set-algorithm and needs a decision-maker to return the most convenient maneuver, whereas the second one requires a space decomposition before fitting a path or a trajectory, and is thus a solve-algorithm. Both are considered for the generation function.

The point-free curves subfamily refers to the principle of lattice, called tentacles algorithm, as introduced in [Von Hundelshausen *et al.* 2008]; see Figure 2.9a. Instead of a space decomposition, the tentacles are based on primitives parametric curves, such as lines and circles, clothoids, and sigmoids. Each tentacle is obtained with different lateral, i.e. steering wheel angle, and/or longitudinal, i.e. speed, parameters. Moreover, as the search space for solutions is reduced, the computational time is limited compared to the space decomposition methods. The tentacles can also be calculated offline, as a trajectory data base [Cherubini *et al.* 2012].

Reference [Dubins 1957] shows that **line and circle** paths are the solution for curves

of minimal length with constraints on curvature and start/end positions. In [Vanholme *et al.* 2013], the authors work with a lane-based zone model built on the linear minimum and maximum trajectories. Those curve configurations are tested on tracks in [Villagra *et al.* 2012, Cherubini *et al.* 2012]. Despite the simplicity and good behavior with high curvature, the second order of line and circle curves is not continuous and hence not realistic for the curvature continuity of the vehicle model. Regarding the road design, lines and circles are linked with **clothoid** functions to obtain a continuous curvature function. Indeed, the clothoid has its curvature proportional to the curvilinear abscissa [Mouhagir *et al.* 2016]. This condition is important for the limitation of the lateral acceleration and thus the vehicle's comfort [Lima *et al.* 2015b]. For example, [Chebly *et al.* 2015] selects clothoid tentacles for overtaking trajectories based on clearance, change of curvature, and trajectory orientation criteria. The authors in [Sheckells *et al.* 2017] propose a fast method to generate a piecewise clothoid curve in agreement with a reference line (e.g. ego lane centerline for highway planning), and kinematic constraints. Conversely, [Lima *et al.* 2015a] focuses on clothoid path sparsification to perform better optimization. However, the clothoid presents an iterative construction process, which increases its calculation time.

The use of straight lines, curves, and arcs based on clothoids paths are also favored to generate reference trajectories in a road-aligned coordinate system [Hudecek & Eckstein 2014]. Nevertheless, the highway curvature is usually small enough for acceptable approximation of a straight road. Under this assumption, only two path's geometries exist for the vehicle: going straight with a straight line or changing lane. For the second case, the **sigmoid** or S-function appears to be an easy solution [Arbitmann *et al.* 2012]. The authors in [Claussmann *et al.* 2015] use sigmoids to generate different candidate trajectories with acceleration profiles based on experiments from different driver behaviors.

The point-based curves subfamily is well suited to geometrically constrained environments and to ensure that the dynamic constraints of the ego vehicle are respected. The principle is to determine control points in the environment, and to fit them with a curve. They can also be used for a smoothing step in other motion planning algorithms. In [Gu *et al.* 2013], the authors use a cubic spiral for path generation, as do [McNaughton *et al.* 2011], and a cubic function of time for velocity generation, whereas in [Tehrani *et al.* 2015] quartic and quintic polynomials are respectively used to generate longitudinal and lateral motions. In [Xu *et al.* 2012], the authors compare cubic **polynomial functions** and smoother quartic curvature polynomials to generate the path with cubic polynomial speed profiles, based on a sampling point lattice.

The use of semi-parametric **spline curves** is an improvement on polynomial interpolations, which are difficult to find and increase in complexity. This entails defining curves as a set of piecewise polynomials. In this way, the obtained polynomial equations are of a lower degree but there are a large number of polynomials to deal with. For example, [Ziegler & Stiller 2009] generates quintic spline trajectories adapted to the road shape. The authors in [Schubert *et al.* 2008] develop adaptive polar splines with non-zero curvature at the beginning and end segments to suit highway maneuvers. Among spline curves, we distinguish **Bézier curves**; see Figure 2.9b. They use control points instead of interpolation points as inflection points. The inconvenient is thus not to pass trough the defined control points, except for the start and end points. The advantages are their simple implementation and thus a low computation cost compared to the previously discussed methods. Reference [Choi 2014] smooths

the primitive path from three waypoints with a quadratic Bézier curve to guarantee kinematic feasibility and avoid static obstacles. In [Chen *et al.* 2013], the authors also use piecewise quadratic Bézier curves based on safe lane change distances for autonomous vehicles and ride comfort. It is also possible to provide the control points of the Bézier curve from an SVM (see subsection 2.2.2.6), as in [Huy *et al.* 2013], which calculates three 4th-degree Bézier curves.

As previously mentioned, the curve algorithms are largely used to interface with all others motion planning algorithms. The choice of the curve type mostly depends on the type of problem and the knowledge of the environment.

2.2.2.5 Numerical optimization

The optimization problem for motion planning is defined as a solve-algorithm based on logic and heuristic approaches. They are part of decision and generation functions. The optimization is usually expressed as the minimization of a cost function in a sequence of states variables under a set of constraints, and is part of competitive combinatorial operational research to avoid a combinatorial explosion. In motion planning applications, we distinguish two domains of interest, as described in [Hart *et al.* 1968]. The first one focuses on finding efficient algorithms to solve complex problems and to improve search time with a heuristic approach (see subsection 2.2.2.2 for some heuristics details). The second one is the mathematical study of the problem to deduce particular properties to find a predictive solution in a restrictive space. As the first domain is commonly used to decrease the computation time of algorithms, only the second domain is further discussed in the following.

The considered approaches model the ego vehicle and environment constraints in a well-defined mathematic form. The main advantages are that they easily handle the constraints of the problem, they deal with multi-criteria optimization, and they consider the state dimensionality and kinematics of the vehicle model [Altché *et al.* 2017]. The basic resolution is the **Linear Programming** (LP). In LP formulation, the algorithm solves a linear cost function under linear equalities or inequalities. The Simplex algorithm is one of the most popular ones; see [Xu *et al.* 2012]. In [Plessen *et al.* 2017], a spatially based trajectory planning with Sequential LP (SLP) is proposed. For nonlinear problems (NLP), nonlinear optimization is used either in the special case of nonlinear regression problems, such as the Levenberg-Marquardt algorithm on path optimization in [Gu *et al.* 2013, Keller *et al.* 2014], or in nonlinear integration, as explained in [Li *et al.* 2014] with a Boundary Value Problem (BVP) solver. For multi-objective problems, the use of **Quadratic Programming** optimization (QP) involves an iterative search of a convex approximation solution to the original problem, as in [Ziegler *et al.* 2014] with sequential quadratic programming (SQP) optimization on distance offset, velocity quadratic error, acceleration, jerk, and yaw rate functions. In [Qian *et al.* 2016, Miller *et al.* 2018], the authors formalize the problem of lane changing and overtaking with a Mixed Integer Quadratic Programming (MIQP).

For specific predictive applications, resolution under **Model Predictive Control** (MPC) is highly popular [Liu *et al.* 2017a, Nilsson *et al.* 2017, Wang & Ayalew 2016, Lima *et al.* 2015a, Altché *et al.* 2017, Lefevre *et al.* 2016, Cardoso *et al.* 2017]. MPC algorithms solve the problem at each sampling time to find a predictive motion solution over a longer horizon time, but only apply the first sequence of actions. In that respect, MPC models a receding horizon control and shifts the solution set to remain accurate to upcoming information. The main advantage

of MPC algorithms is their replanning ability, but they are still too poor for non-convex and high complexity problems.

Dynamic Programming (DP) draws its efficacy for complex computational problems by breaking them into simpler subproblems, even with interdependency. The resolution of each subproblem is combined to find the global problem solution. This implies that the description of the problems has good characteristics. In [Choi 2014], the author searches for the shortest path in a cells decomposition using a pre-calculated DP solution. The authors in [Gu *et al.* 2016a] use DP to calculate the optimal cost-to-go value of a set of maneuvers for different speed profiles.

Numerical optimization is widely used in motion planning, either to decrease the solving time of a graph's exploration, or to exploit the mathematical properties of the problem. These algorithms can be solved by generic numerical resolution tools. The main encounter frameworks in the literature are CVX and CVXGEN softwares, Gurobi and YALMIP solvers, Matlab Optimization Toolbox, NPSOL package, and the ACADO Toolkit.

2.2.2.6 Artificial intelligence

The main contribution of Artificial Intelligence (AI) for autonomous driving is their ability to reproduce and simulate drivers' reasoning and learning. These techniques rely on thinking and acting consistently with the environment, a memory structure, and drawing inferences. In this sense, AI algorithms are particularly interesting for the decision making function. They are also well suited to mobile robotics, as they are flexible, adaptive, and reactive to their environment. Moreover, AI techniques are well organized to deal with huge, incomplete, or inaccurate data. The advantages of AI-based algorithms are their capacity to answer generic questions and to absorb new modifications without affecting the structure of the algorithm. They are mostly employed as solve-algorithms for predictive planning, but also as symbolic set-algorithms, and less frequently for reactive deformation. AI gathers a wide diversity of methods from logic to cognitive representation. We propose to organize this section in two main axes – cognitive/rational and rules/learning distinctions – based on [Russell *et al.* 1995]'s distinction between thinking and acting humanly or rationally. We thus distinguish the four subfamilies depicted in Figure 2.10: logic approaches, heuristic algorithms, approximate reasoning, and human-like methods.

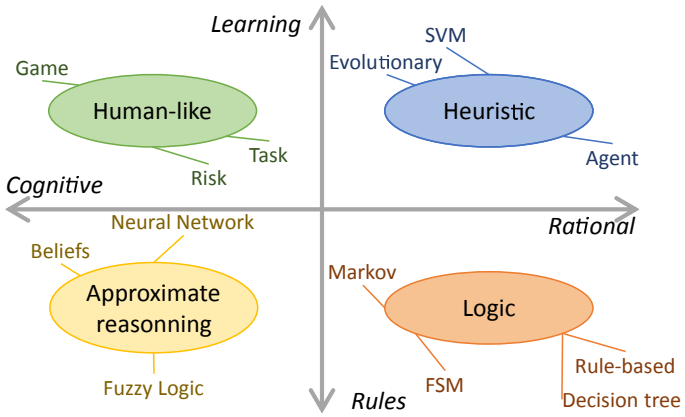


Figure 2.10: A map of AI-based algorithms.

AI Logic-Based Approach AI logic-based approaches are symbolic planning used in decision making. They define expert systems to solve specific complex tasks depending on a knowledge base with an inference engine to automate the reasoning system. In case of modification, addition, or removal in the knowledge base, the inference engine should also be updated, and a recursive mechanism must be applied to guarantee the convergence of the new expert system. AI logic methods serve as set- and solve algorithms, while generating or selecting a set of time/space states or actions. Furthermore, their fast architecture allows their use for predictive or reactive planning. The main advantage of these systems is their intuitive setup to emulate human logic and rational reasoning. On the other hand, the knowledge base requires the discretization of numerous environment variables with a high number of cause-and-effect rules and tuning parameters.

The best-known inference engine is **rule-based** reasoning. The statements are if observations, then actions. The authors in [Nilsson *et al.* 2015] report satisfying results of rule-based intention prediction on highways, and use it to perform lane change maneuvers in [Nilsson *et al.* 2016]. The major advantages are to clearly identify the cause and effects, the notational convenience and the straightforward implementation. The main drawbacks are the cyclic reasoning and the exhaustive enumeration of rules, which lead to infinite loop and impact the computation time. Furthermore, if the current situation does not match the observations in the rule base, an unsuitable default decision may be made. In such a case, the knowledge data should be enriched and modified offline. Moreover, as a declarative reasoning, the rules' order matters and resolution conflicts can happen. A solution would be to add an heuristic to prioritize the rules.

To display rule mechanisms, **decision trees** are promoted as compressed graphical representations and decision support tools. In [Constantin *et al.* 2014], a decision-tree is depicted by enumerating all the possible navigation lanes. To facilitate the organization of the tree, binary decision diagrams or flowcharts are developed to represent Boolean functions, as applied in [Claussmann *et al.* 2015, Li *et al.* 2017]. While decision trees are simple to interpret, however, calculations become highly complex with uncertain or approximate values. On the other hand, decision rules must be interpreted to ensure safe behavior and to detect and anticipate non-legal and dangerous behavior of other vehicles, as developed in [Vanholme *et al.* 2013] with the concept of legal safety.

To avoid the exhaustive rules declaration, the **Finite State Machine** (FSM) gives an abstract model of the system behavior, representing the system states linked by actions/conditions. In [Ardelt *et al.* 2010], the FSM separately describes two longitudinal and four lateral state transitions with a specific contribution to an emergency stop assistant on highways. Compared to the rule-based approach, FSMs directly perform a predetermined sequence of actions and states, which are then mapped with path generation and control, as done in [Wang *et al.* 2015]; see Figure 2.11. They can also be considered as state classifier algorithms [Tehrani *et al.* 2015], and thus represent an easy communication tool for collective and driver-shared driving. FSMs are not imperatively deterministic, which allows for more complex states' relationships. Reference [Ziegler *et al.* 2014] exploits multiple state charts in parallel to deal with concurrent states, which are well suited to performing simultaneous actions in a decision process (e.g. yielding and merging). The main disadvantage of the previously discussed FSM representations is that they are only based on certainty in knowledge and can not be generalized to unknown situations.

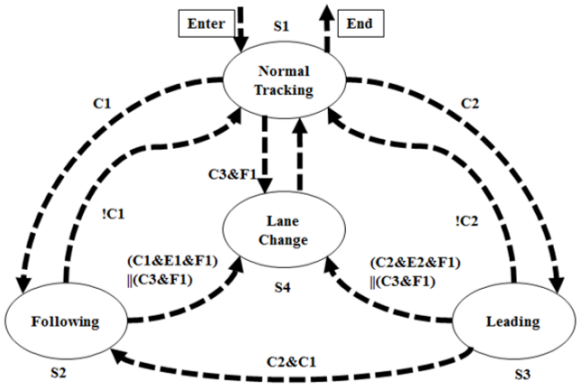


Figure 2.11: Illustration of an FSM for highway [Wang *et al.* 2015]. C_i, E_i, F_i are specific conditions to transit between the vehicle behaviors S_i.

In case of uncertainty, **Bayesian networks** using Markov models are employed. They are statistical representations of causal links, based on probabilistic transitions. They present a knowledge-based identification step of their parameters to determine the most likely sequence of states to the sequence of outputs. Reference [Rodrigues *et al.* 2016] matches the most expected obstacles' behavior intention to return the best ego behavior maneuver (stop | cruise | accelerate | decelerate and turn-right | turn-left | straight). The authors in [Mouhagir *et al.* 2016] develop a **Markov Decision Process** (MDP) on the choice of tentacle trajectories, and the one in [Zhou *et al.* 2017] for a lane-staying or -changing decision. The advantage of MDP is their ability to evaluate several predictions at the same time. Behavior improvements are shown using a Partially Observable Markov Decision Process (POMDP) with a probability distribution over the set of possible states in [Ulbrich & Maurer 2015, Galceran *et al.* 2017, Li *et al.* 2018].

The AI logic-based methods are mostly suitable in constrained and predictable environments, such as nominal highway driving. Their intuitive and fast architectures are widely promoted for use in critical safety environments, where cause-and-effect reasoning is necessary. On the other hand, their lack of autonomy and rigid program structure for an adaptable and reconfigurable algorithm are their main disadvantages for use in open environments such as highways. As the logic-based approaches are straightforward to implement, no specific framework stands out in the literature.

AI Heuristic Algorithms Heuristic algorithms are experience-based and conducive to natural environment process exploration. They aim to find an approximate solution, and are therefore used as faster and more efficient algorithms when the traditional exhaustive methods fail. In motion planning, they usually return a set of actions, but are also able to return paths or trajectories and to act as set- and solve- algorithm in predictive or reactive planning. Their main advantage is their low computational time and complexity, and their ability to handle complex problems. On the other hand, they provide, by definition, a local solution, which does not guarantee global optimality and accuracy.

The most convenient interpretation of heuristic methods involves an **agent** representation. An intelligent agent is an autonomous entity modeled with a rational and social behavior, which adapts to the observations of the environment. Its behavior can be based on condition-

action rules (only depending on the current perception), a world model (how it affects the environment), goals to achieve, or utility to goals, e.g on a game theoretic formalism in [Li *et al.* 2018]. The faculties of heuristic agents allow the inclusion of multipolicy decision making, such as in [Galceran *et al.* 2017], including distance to goal, lane choice bias, maximum yaw rate, and policy cost. They are especially well suited to uncertain environments. The main drawback is the difficulty of ensuring convergence towards the solution.

Learning methods are also introduced to the decision in heuristic approaches. **Support Vector Machines** (SVM) are statistical learning classifiers for agent intentions which depend on information search algorithms. In analogy with an FSM, they are based on states' classes and in-between margins. The separation of the classes has to be trained beforehand, and full labeling of input data is needed to return a convenient classification. In [Huy *et al.* 2013], the authors develop an SVM to provide the control points to a Bézier curve-fitting method. Authors in [Vallon *et al.* 2017] define an SVM for personalized lane change decision based on the relative velocities and positions.

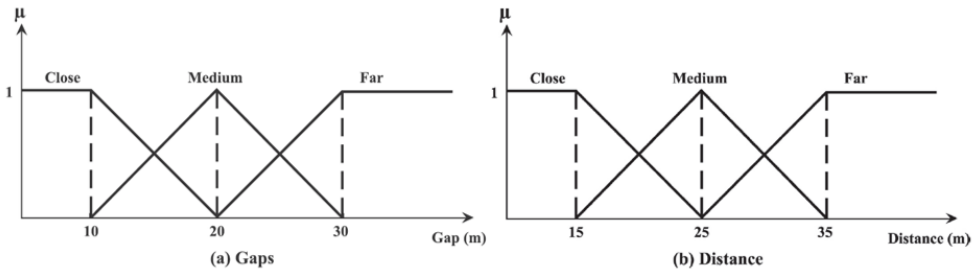
Evolutionary methods are a more widely used class of learning algorithms. They are defined as meta-heuristic functions, inspired by biomimetics with a natural rational evolution process, such as reproduction, mutation, recombination, and selection. The first step is to determine a set of a priori solutions associated with a fitness function to evaluate their quality. A set of evolution processes is then applied to find a better solution to the optimization problem. In [Baluja *et al.* 1997], the authors develop the reasoning system SAPIENT, based on a population-based incremental learning, to define the most appropriate parameters for a given task to solve tactical driving problems. More recently, [Onieva *et al.* 2011] applies two genetic algorithms to refine a fuzzy control module, and [Riaz *et al.* 2015] details a genetic algorithm with a 4-chromosome structure based on speed, angle, break and time to avoidance. The risk of such algorithms is linked to the mutation mechanism, whose random process can lead to local minima. They are of particular interest for agent swarm methods.

AI heuristic algorithms are a good alternative to address the disadvantages of classical methods. Their attributes are acceptable for highway driving, where an optimal solution is not necessary and approximate solutions could be sufficient. There is no AI-heuristic framework built for the autonomous vehicle applications, however, major frameworks are available to solve this family, e.g. JADE for agents, DEAP, Jenetics or the Matlab Global Optimization Toolbox for evolutionary algorithms.

AI Approximate Reasoning AI approximate reasoning mimics human reasoning. We distinguish the logic approach, as described earlier with expert systems, with the difference that the knowledge base is non-Boolean; and the learning approach to classify new knowledge to adapt to future situations. The first presents the advantage of exhibiting intelligent behavior with intuitive demonstration and explanation, whereas the second is related to a system able to understand, think, and learn.

Compared to Boolean logic, **fuzzy logic** relies on many-valued variables. It consists of fuzzy expert systems based on Boolean compromises, which mathematically model vagueness and are close to cognitive reasoning with an inductive logic programming on different attribute-level solutions. They return a decision, usually expressed as a maneuver or task, and are thus part of predictive or reactive solve-algorithms. Their major advantages are the flexibility and permissiveness of the designed rules and by extension to uncertain data.

On the other hand, their main drawbacks are their lack of traceability and the absence of a systematic design methodology. The authors in [Balal *et al.* 2016] propose a fuzzy inference decision system to check whether the ego vehicle has enough time to change lanes based on a close, medium, or far evaluation of the gaps between the surrounding vehicles; see Figure 2.12. References [Naranjo *et al.* 2008] and [Milanés *et al.* 2011] use fuzzy rules on lateral and angular error, respectively, to propose automatic overtaking maneuvers and lateral following of a reference map.



Rule no.	IF				THEN
	$\widetilde{G}_{FA}$	$\widetilde{G}_{PA}$	$\widetilde{D}$	$\widetilde{G}_{PB}$	$\widetilde{C}$
1	Close	Close	Close	Close	No
2	Close	Medium	Close	Close	Yes
3	Close	Close	Medium	Close	Yes
⋮					
49	Far	Medium	Medium	Close	Yes
50	Far	Close	Close	Close	Yes
51	Far	Close	Close	Medium	Yes

Figure 2.12: Illustration of the fuzzy logic decision in [Balal *et al.* 2016].

Artificial Neural Networks (ANN) are a popular form of approximate learning solutions. ANNs imitate the low-level structure of biological neural networks, i.e. connected neurons in overlaid layers. The learning process is based on error retropropagation to adjust the neural connections' weights with different learning strategies, e.g. supervised, unsupervised, reinforcement learning. They are used for both generation and decision making functions, as a set- and solve-algorithm in predictive and reactive approaches. Their main drawback for autonomous driving is the absence of a causal explanation to a solution, while their main advantage is their ability to learn by training on multidimensional data. Another disadvantage is thus the large amount of data required, which makes them computationally expensive. Reference [Lefevre *et al.* 2016] proposes a combination of Hidden Markov Models (HMM) and Gaussian Mixture Regression to mimic the driver's acceleration and braking model, whereas [Geng *et al.* 2016] learns the driving speed profiles with a Levenberg-Marquardt algorithm on a 3-layer ANN with 20 hidden nodes trained on 600 driving trips. The authors in [Rehder *et al.* 2017] imitate driver trajectories for lane changing with **Convolutional Neural Networks** (CNN), and [Bojarski *et al.* 2016] compares the learning stage with human driver control inputs. **Reinforcement learning** is used in [Li *et al.* 2018] to determine the policies for each agent in the scene representation, or for multi-goal overtaking maneuvers in [Ngai & Yung 2011] and automated lane change maneuvers in [Wang *et al.* 2018a]. Some new promising research is in progress using the deep learning methods [Chen *et al.* 2015b], or by integrating the perception and planning blocks in **end-to-end learning** [Yang *et al.* 2018].

Neural networks are also extended in case of uncertainty with Bayesian probabilities in **belief networks** [Huang *et al.* 1994]. However, one of the challenges of these methods is ensuring that they learn in the correct way with the diversity of environments and behaviors of driving.

AI approximate reasoning algorithms seem to be highly promising for the near future. They rely on logic and statistical bases, which are extended to cognitive properties. They also provide adaptive reasoning to evolve appropriately to their environment. At the moment, the main impediment to these algorithms is the lack of feedback in real driving scenarios. As the real-time implementation of approximate reasoning algorithms is complex, the use of specific frameworks is favored, e.g. Matlab Fuzzy Logic Toolbox for fuzzy logic approach or TensorFlow for machine learning.

AI Human-Like Methods AI human-like methods propose high-level models that mirror human processes for solve-algorithms. We do not pretend to provide a complete description of driver models; for this, the readers are invited to refer to [Salvucci 2006]. The AI human-like methods discussed in this thesis are decision making functions, based on human cognitive aspects. They are intelligent, potentially learning, and cognitive procedures. Their knowledge of the environment rules and ability to handle various and complex situations make them useful for predictive and reactive planning. Their main advantage is an abstract and universal representation of the decision making process. However, they are difficult to model and to use accurately. In [Michon 1985], the author highlights different driver models based on taxonomic or functional, and behavioral or psychological distinctions. With respect to his decomposition, we categorize AI human-like methods into three main approaches based on risk, task, and game theories.

Risk estimators are employed to interpret rational decisions from the scene analysis with a cognitive bias. The main advantage is to be intuitive. This implies a balance between a subjective level of acceptable risk and the objective safety, which is a negative aspect for a global risk assessment. The authors in [Lefèvre *et al.* 2014] identify two notions to evaluate the risk of a situation: risk of collision and risky unexpected behavior. In the scope of our work, we only details algorithms for the risk of collision. The most basic risk estimator uses a binary collision prediction. Considering a risk criterion, a threshold is applied to the obtained value to classify it as risky or safe. Indicators such as **Time-To-Collision** (TTC) [Wang & Ayalew 2016, Ardelt *et al.* 2010, Glaser *et al.* 2010] or **Time-To-React** (TTR) [Tamke *et al.* 2011, Wagner *et al.* 2018], e.g. steering or braking, give measurable estimators to the driver too. This method is also exploited in [Bautin *et al.* 2010] to return Inevitable Collision States (ICS) for grid occupation representation (see subsection 2.2.2.1). The **Time-To-Intercept** (TTI) indicator is extended in [Ghumman *et al.* 2008] to return a position-velocity shadow target to prevent obstacle collision. Binary risk estimators provide a coarse evaluation of the scene, but uncertainty of the obstacles' future motion is a parameter not to neglect. Probabilistic risk estimators are better suited to providing a more realistic scene representation.

To adapt the risk perception to the driver, **compensation risk** estimators introduce driver states such as stress, drowsiness, or illness. In [Taylor 1964], the authors use a risk-speed compensation model, i.e. the product of perceived risk and the driver's speed is constant. Reference [Wilde 1982] exploits a risk homeostasis theory to deal with uncertainties over subjective risk assessment. Furthermore, the authors in [Naumann & Stiller 2017] in-

troduce a **Time of Zone Clearance** (TZC), combined with the perceived risk on speed, distance, safety, and comfort, and based on the proposition that risk appears if trajectories overlap.

Even if compensation risk estimators seem to be more faithful to the human's decision, the above risk approaches are usually based on one factor of risk, whereas risk warning should take into account all factors of the scene representation. Reference [Prokhorov 2009] proposes a two-level risk estimator affected by weather, traffic, or road conditions, and then refined with real-time information about the ego vehicle surroundings. Promising risk assessments have been developed combining usual risk estimators and belief theory, as proposed in [Daniel *et al.* 2013] to relax thresholds applied to fuzzy sets.

Task 32: Passing

```
32-1    Decides wether to pass (two- or three-lane roads)

32-11   Looks along roadside for no passing control signs
32-111   Does no pass if « no passing » zone is indicated or has been
         indicated previously
32-112   May pass if sign indicates end of « no passing » zone

32-12   Observes lane markings
32-121   Does not pass if left side of lane is marked by the following:
32-1211   One or two solid lines
32-1212   Solid line to the right of broken line
```

Figure 2.13: Illustration of the first elementary tasks for "passing" in [McKnight & Adams 1970a].

The second approach comprises **taxonomic models**, which are also popular symbolic models for human decisions. They identify the sequential driving tasks relations, with behavioral and ability requirements. Those methods call for a logical organization, and represent operational research task scheduling. They must then be interpreted either as space-time-action constraints or with a local planner, e.g. the curve planner discussed in subsection 2.2.2.4. One of the disadvantages is their difficult explicit description, as they are either focused on a driving task, such as lane changing or merging, or hardly exhaustive. However, they present the advantages of providing an abstract and universal representation and easy replanning. For example, McKnight and Adams list 45 major driving tasks in [McKnight & Adams 1970b, McKnight & Adams 1970a, McKnight & Hundt 1971b, McKnight & Hundt 1971a], decomposed into 1700 elementary tasks; see Figure 2.13. References [Naranjo *et al.* 2008, Chen *et al.* 2015a] define the sequence of operations to perform an overtaking maneuver; each action is then validated under a set of numerical criteria to prompt the next action. The authors in [Bahram *et al.* 2014] propose a functional architecture of the driving strategy as a discrete set of behavioral strategies for a specific traffic situation.

Third, some learning approaches are inspired by interaction models introduced in **game theory**. The idea for highway planning is to consider the moving vehicles as players that observe each other's actions and consequently react with an appropriate strategy. The main advantage of this method is that it quickly obtains a trained driver model, starting with a merely basic one. The main drawback is that it assumes that all the players respect the rules, and it can therefore lead to unsafe reactions in real-life applications. Moreover, one should make sure to learn using various behaviors of the players to enrich the knowledge. In [Li *et al.* 2018], the models develop more complex strategies as they are trained against the other behaviors. The authors use POMDP to model the players' knowledge and Jaakola

reinforcement learning in the training phase.

AI human-like methods are well suited to decision making in highway scenarios, where drivers' behaviors are more predictable due to the basic rules of this environment. They are also easy to understand and to share with the driver. Moreover, the application of such algorithms is usually not as complicated as modeling a driver, but still interesting enough to involve in complex scenarios. With their simple architecture and heterogeneous implementation, one notices that no major framework is highlighted in the literature of autonomous driving.

2.3 Comparison table for highway applications

The highway applicability of the previously described methods is summarized in Tables 2.2 and 2.3. We propose to quantify the constraints' assessment with a '– / ∼ / +' scale. We assess '– –' as being highly inappropriate, '–' inappropriate, '∼' intermediary, '+' appropriate, and '++' highly appropriate in Table 2.2. The references are gathered in Table 2.3, and illustrate the use cases of the families to suggest their situation adaptiveness (collision avoidance, car following, lane change, merging, overtaking) and their implementation in simulation or experimentation, specified with the square brackets' superscript and subscript respectively. To quote but a few, simulators such as CARLA, PreScan, SCANeR studio, TORCS are developed for autonomous driving.

Unlike the classification in [Paden *et al.* 2016], we do not focus on the usual points of comparison, such as completeness, optimality, or time complexity, but on how the algorithms offer an effective and efficiently implementable response to practical applications. Therefore, one can interpret Table 2.2 as a guide to choose the most appropriate family given the attributes of the algorithm, and its intrinsic and extrinsic limits. Furthermore, one will notice that references often apply to different families. In that sense, Table 2.2 also helps to understand the complementary methods for a systemic motion planner design.

Among the characteristics highlighted, the taxonomy criteria can be found in subsection 2.2.1: (i) the **type of use** of set- and solve-algorithm, (ii) the predictive or reactive **horizon**, and (iii) the **mathematical domain**; and the motion planning **functions** in subsection 2.1.2 (Figure 2.3).

As intrinsic limits, we distinguish the performance, ease of use, and data analysis. The performance first gathers the **real-time** implementation. In the case of highway motion ($> 17m/s$), with an error of a meter, the motion planning algorithm should return a solution within the order of magnitude of $10ms$ to be real-time. As the intrinsic cost of each algorithm strongly relies on the hardware platform, we inform about qualitative, but not quantitative computation. The other performance requirements are derived from the analysis proposed in [Khatib & Le Maitre 1978]: **robustness** of the algorithm to find a solution despite the variation of the environment (merging, jam approach); **stability** to keep the solution despite environmental changes; **adaptability** to perform a solution in various conditions, e.g. introducing new scenarios or criteria; and the **feasibility** of converging to a solution in a finite time. The ease of use is defined by the ability to **replan** in real-time without changing the structure of the algorithm. The last intrinsic limit relates to the data analysis: the **input type** for the scene representation (discrete, sampled, continuous) and the **output type** of

the algorithm (space, path, trajectory, maneuver, task), along with the ability to deal with **uncertain** data.

The extrinsic limits show the algorithm's dependency on high-level sensors of the ego vehicle and/or the infrastructure. This aspect is important to reflect the real applicability in a near or further future. If the algorithm works well only with precise information over a large time horizon, which does not correspond to the current high-level sensors' properties, it will be highly inappropriate for **sensors constraints**. We invite the readers to consider [Gruyer *et al.* 2017] as a reference for sensors' properties. As stated before, the specificities of the driving environment and vehicle kinematics must be considered. We therefore consider whether the algorithm performs well in **complex environments** (dense traffic, various topologies) and takes the **constraints of the environment** and the **ego vehicle kinematics** into account.

2.4 Conclusion and positioning of the research

To conclude, our literature review revealed a considerable amount of algorithms for motion planning in robotics. The objective of this chapter was to identify the main methods of motion planning for autonomous vehicles in the context of highway driving, in order to propose new research contributions. We do not claim to have exhaustively gathered all algorithms used for this application, but we hope to have shown the diversity and potential of highway planning. In addition, we propose a novel decomposition of the methods for the state of the art. In particular, we believe that the radar chart illustration of differentiating attributes will help future research to identify and orient the work towards solutions that will best address the problem.

From this literature review of motion planning algorithms, we were able to identify different areas of improvement to study for the positioning of our research.

The first area consists in the improvement of the planning architecture, in particular, the relationship between generation and decision making. In previous works, the functions of motion generation and decision making are addressed separately or in a sequential way, applying decision making to decide on a maneuver and then motion generation to properly fit this maneuver, or vice versa. However, this separation may lead to a final motion that will not be consistent or optimal with respect to the navigation environment and associated criteria. For example, during an overtaking maneuver, the return lane change may not be optimal if there is another obstacle on the arrival lane. In this case, it may be better to generate a motion to stay on the passing lane for a second passing. We therefore propose a new combined architecture for decision making and motion generation so that both functions are optimized in relation to each other. The advantage of this architecture is also its ability to work in a continuous solution space (no need for discretization). Similar combined approaches also emerge within the research in perception that uses learning methods. Nevertheless, one of the major drawbacks of these learning algorithms is the lack of cause and effect explanations and reproducibility. In fact, they are not acceptable for decision making that involves human safety. Therefore, we will present a deterministic decision-maker, as well as a deterministic characterization of the evolution space.

A second area of improvement concerns the workspace in the context of motion generation.

Indeed, many approaches rely on a discretization of the evolution space. This is the case for methods of space configuration by sampling, cells decomposition, or pathfinding algorithms. Other methods consider an incomplete evolution space by selecting predefined paths, such as lattice or tentacle approaches. By doing so, these methods do not allow the optimal motion to be found since the evolution space is truncated. Continuous space approaches such as homotopy or field methods have a major disadvantage in terms of computation time and are therefore forced to be used as a reactive rather than a predictive approach to be embedded in real time on the vehicle. To address the workspace problem, we consider the navigable space as a set of intervals. These intervals are previously reduced to a collision-free space, i.e. the vehicle is in a non-conflict situation. The distinction between intervals is made by road lanes and spaces between obstacles. Finally, the high-level maneuvers are carried out to move from one interval to another. It should also be noted that the interval delimits a continuous space.

Moreover, the dynamic trajectory is often optimized separately from the static path, as in curves approaches. This is why we propose to work on the intervals previously discussed in space and time as a third area of improvement. The prediction of obstacles and the dynamically feasible motions of the vehicle ego characterize the spatiotemporal intervals.

Finally, with regard to the decision making or selection part, the main drawback of existing methods concerns the multi-criteria approach, while taking into account uncertainty and trustworthiness. Many methods correctly manage a multi-criteria decision but it is then necessary to deal with homogeneous criteria or in limited number for reasonable calculation time and to avoid combinatorial explosion, as in the case of pathfinding algorithms. Logic algorithms or risk approaches may be too rigid for real life cases and quite complex in managing information from different sources. In addition, it is most often a matter of setting acceptance thresholds, which results in a loss of information for decision making. This fourth area of improvement is thus addressed by using fuzzy logic to easily combine heterogeneous criteria and integrating it into a belief theory framework. The advantage of this framework is to assign criteria into categories corresponding to different decision-states, such as safety around the vehicle, the dynamic limits of the vehicle, passenger comfort or compliance with driving rules. The categories are then considered as different sources of information to obtain the final decision. Moreover, the consideration of uncertainties and trustworthiness becomes possible by using fuzzy logic and the weighting of criteria or categories. This framework also makes it easy to adapt the vehicle's behavior to the driving style. Nevertheless, the resulting function for decision making is no longer explicit. It is then impossible to use methods of numerical optimization from the state of the art. To overcome this, we consider the advantage of heuristic approaches that allow us to find a near-optimal solution in a finite time. Besides, it becomes possible to adjust the calculation time according to the encountered situation, while guaranteeing to return an acceptable solution.

These identified areas of improvement are explored in the next two chapters. Chapter 3 will detail the contribution to the decision making function. The planning architecture as well as the space-time evolution space are broached in Chapter 4.

Table 2.2: Comparison table for highway applications of motion planning methods ('−−' very inappropriate, '−' inappropriate,'∼' intermediary, '+' appropriate, '++' very appropriate).

Family	Characteristics				Intrinsic Limits									Extrinsic Limits			
					Performance					Use	Data analysis			Sensor	Environment		Vehicle
	Type of use	Horizon	Domain	Functions	Real-time	Robustness	Stability	Adaptability	Feasibility	Replanning	Input Type	Output type	Uncertainty	Constraints	Complexity	Highway constraints	Kinematics
Configuration space sampling	se	P/R	G	gen	∼	−	∼	∼	−	++	s	sp	−	∼	−	+	++
Probabilistic sampling	se	P/R	G	gen	+	−−	−−	+	−−	++	s	sp	+	+	∼	−−	−−
Non-obstacle-based cells																	
– Exact decomposition; Polar grid	se	P/R	G	gen	−	∼	−	∼	∼	−	d	sp/pa	∼	++	−	−−	−−
– VFH ; DW	se	R	G	def	+	++	∼	+	+	+	c	sp/tr	+	−	+	+	∼
Obstacle-based cells																	
– Voronoi ; Approximate	se	P/R	G	gen	−−	++	∼	+	∼	−−	d	sp/pa	+	−	−	+	−
– Visibility graph	se	P/R	G	gen	−	∼	−−	−	∼	−−	d	sp/pa	−	∼	−−	∼	−−
– VO ; Driving corridor	se	P/R	G	gen	−	++	∼	+	+	−	c	sp/tr	∼	−	∼	+	+
Lattice																	
– Maximum turn; Curvature Velocity	se	P	G	gen	∼	+	+	+	+	+	c	tr/ma	−−	+	−	+	++
– Road-adapted; Driver-based	se	P	G	gen	∼	+	+	+	+	+	c	tr/ma	−	∼	−	++	+
Pathfindings																	
– Dijkstra	so	P	L	dec	−	∼	+	+	++	∼	s	pa	−	+	−−	−−	−−
– A*	so	P	H/L	dec	∼	∼	+	+	+	+	s	pa	∼	+	−	∼	−−
– RRT	se/so	P/R	H/L	gen/dec	+	+	∼	++	∼	++	s	pa	+	−	−	∼	++
Artificial forces																	
APF; VFF; Elastic band	se/so	R	B	gen/def	−	+	+	+	+	++	c	sp/pa/tr	+	+	∼	++	+
Parametric curves																	
– Line and circle	se	P/R	G	gen	+	−	−−	+	+	+	s/d/c	pa/tr/ma	−−	+	−−	+	∼
– Clothoïd; Sigmoid	se	P/R	G	gen	+	−	−−	+	+	+	s/d/c	pa/tr/ma	−	+	∼	++	++
Semi-parametric curves																	
Polynomial; Spline; Bézier	so	P/R	G	gen	∼	−	−−	+	++	+	s/d/c	pa/tr/ma	∼	+	+	+	++

Legend:
Characteristics – se/so = set-/solve-algorithm; P = predictive / R = reactive; G = geometric / H = heuristic / L = logic / C = cognitive / B = biomimetic; gen = generation / dec = decision making / def = deformation.
Data Analysis – d = discrete / s = sampled / c = continuous; sp = space / pa = path / tr = trajectory / ma = maneuver / ta = task

Table 2.2: (continued)

Family	Characteristics				Intrinsic Limits									Extrinsic Limits			
					Performance					Use	Data analysis			Sensor	Environment		Vehicle
	Type of use	Horizon	Domain	Functions	Real-time	Robustness	Stability	Adaptability	Feasibility	Replanning	Input Type	Output type	Uncertainty	Constraints	Complexity	Highway constraints	Kinematics
Mathematical optimization																	
LP; NLP; QP; MPC; DP	so	P	H/L	gen/dec	+	+	+	++	~	+	s/d/c	pa/tr/ma	~	+	+	++	++
AI Logic																	
– if-then-rules	so	P/R	L	dec	~	+	+	–	+	~	d	ta	–	++	–	++	~
– FSM	se/so	P/R	L	dec	+	+	+	–	++	+	d	ta	––	++	~	++	~
– Dynamic Bayesian	se/so	P/R	H/L	dec	–	++	+	–	++	+	d	ta	++	+	~	++	~
AI Heuristic																	
– Agent	se/so	P/R	H/L/B	gen/dec	–	~	–	++	~	+	s/d/c	pa/tr/ma	+	+	+	++	+
– SVM	se/so	P/R	H/L	dec	~	++	~	–	+	––	s/d/c	ma	~	+	+	+	~
– Evolutionary	se/so	P/R	H/L/B	gen/dec	–	++	–	++	––	––	s/d/c	pa/tr/ma	–	+	+	++	+
AI Approximate																	
– Fuzzy logic	so	P/R	H/L/C	dec	~	+	+	~	+	~	c	ma/ta	+	++	~	++	+
– ANN	se/so	P/R	H/L/C/B	gen/dec	––	~	~	––	+	–	c	tr/ma/ta	~	––	+	+	+
– Belief Network	se/so	P/R	H/L/C/B	gen/dec	––	~	~	––	+	–	c	tr/ma/ta	++	–	+	+	+
AI Cognitive																	
– Risk	so	P/R	L/C	dec	++	++	++	+	++	++	d	ta	~	––	+	++	~
– Task	so	P/R	L/C	dec	+	+	~	+	+	–	d	ta	––	++	~	+	~
– Game	se/so	P/R	C	gen/dec	~	~	+	–	++	~	c	ma/ta	+	+	–	+	+

Legend:
Characteristics – se/so = set-/solve-algorithm; P = predictive / R = reactive; G = geometric / H = heuristic / L = logic / C = cognitive / B = biomimetic; gen = generation / dec = decision making / def = deformation.
Data Analysis – d = discrete / s = sampled / c = continuous; sp = space / pa = path / tr = trajectory / ma = maneuver / ta = task

Table 2.3: Use case references table for highway applications of motion planning methods.

Family	Use case
Configuration space sampling	[*Hesseet al.* 2010]$^{o}_{E}$
Probabilistic sampling	[*Villagraet al.* 2012]$^{*}_{E}$ [*Liet al.* 2014]$^{ca}_{S}$
Non-obstacle-based cells	
– Exact decomposition;	[*Ardeltet al.* 2010]$^{e}_{S}$ [*Tehraniet al.* 2015]$^{lc}_{S}$ [*Mouhagiret al.* 2016]$^{o}_{S}$ [*Yuet al.* 2016]$^{*}_{E}$
Polar grid	[*Moraset al.* 2011]$^{*}_{E}$
– VFH;	[*Leeet al.* 2014]$^{*}_{E}$ [*Quet al.* 2015]$^{*}_{S}$
DW	[*de Lima&Pereira* 2013]$^{*}_{E}$ [*Mitschet al.* 2013]$^{*}_{/}$ [*Kanget al.* 2015]$^{ca}_{S}$
Obstacle-based cells	
– Voronoi;	[*Villagraet al.* 2012]$^{*}_{E}$ [*Leeet al.* 2014]$^{*}_{E}$
Approximate	[*Trepagnieret al.* 2011]$^{*}_{E}$
– Visibility graph	[*Johnson&Hauser* 2013]$^{m}_{S}$ [*Choi* 2014]$^{*}_{S}$
– VO;	[*Choi* 2014]$^{*}_{S}$
Driving corridor	[*Zieglaret al.* 2014]$^{*}_{E}$ [*Benderet al.* 2015]$^{*}_{S}$ [*Guet al.* 2016*b*]$^{*}_{S}$ [*Altch&De La Fortelle* 2017]$^{*}_{S}$
Lattice	
– Maximum turn;	[*Schubertet al.* 2008]$^{*}_{S}$ [*Broggiet al.* 2014]$^{*}_{E}$
Curvature Velocity	[*Constantinet al.* 2014]$^{*}_{S}$ [*Guet al.* 2016*b*]$^{*}_{S}$
– Road-adapted;	[*Ziegler&Stiller* 2009]$^{*}_{S}$ [*McNaughtonet al.* 2011]$^{*}_{E}$ [*Xuet al.* 2012]$^{*}_{E}$ [*Guet al.* 2013]$^{*}_{S}$
Driver-based	[*Yaoet al.* 2012]$^{*}_{E}$
Pathfindings	
– Dijkstra	[*Villagraet al.* 2012]$^{*}_{E}$ [*Bahramet al.* 2014]$^{*}_{S}$ [*Guet al.* 2016*b*]$^{*}_{S}$
– A*	[*Rezaeiet al.* 2003]$^{*}_{E}$ [*Dolgovet al.* 2008]$^{*}_{E}$ [*Schubertet al.* 2008]$^{*}_{S}$ [*Hesseet al.* 2010]$^{o}_{E}$ [*Bouniniet al.* 2017]$^{*}_{S}$ [*Boroujeniet al.* 2017]$^{*}_{S}$
– RRT	[*Hwan Jeonet al.* 2013]$^{*}_{S}$ [*Maet al.* 2015]$^{*}_{E}$
Artificial forces	
APF;	[*Reichardt&Shick* 1994]$^{*}_{S}$ [*Gerdes&Rossetter* 2001]$^{*}_{S}$ [*Wolf&Burdick* 2008]$^{*}_{S}$ [*de Lima&Pereira* 2013]$^{*}_{E}$ [*Galceranet al.* 2015]$^{*}_{E}$ [*Bouniniet al.* 2017]$^{*}_{S}$ [*Liuet al.* 2017*a*]$^{*}_{S}$ [*Rasekhipouret al.* 2017]$^{*}_{S}$
VFF;	[*Hesseet al.* 2010]$^{o}_{E}$
Elastic band	[*Gehrig&Stein* 2007]$^{cf}_{E}$ [*Songet al.* 2013]$^{*}_{S}$ [*Kelleret al.* 2014]$^{ca}_{S}$
Parametric curves	
– Line and circle	[*Cherubiniet al.* 2012]$^{ca}_{E}$ [*Villagraet al.* 2012]$^{*}_{E}$ [*Vanholmeet al.* 2013]$^{*}_{E}$
– Clothoïd;	[*Cheblyet al.* 2015]$^{*}_{E}$ [*Limaet al.* 2015*a*]$^{*}_{S}$ [*Limaet al.* 2015*b*]$^{*}_{S}$ [*Mouhagiret al.* 2016]$^{o}_{S}$ [*Sheckellset al.* 2017]$^{*}_{S}$
Sigmoid	[*Arbitmannet al.* 2012]$^{*}_{E}$ [*Claussmannet al.* 2015]$^{lc}_{E}$
Semi-parametric curves	
Polynomial;	[*McNaughtonet al.* 2011]$^{*}_{E}$ [*Xuet al.* 2012]$^{*}_{E}$ [*Guet al.* 2013]$^{*}_{S}$ [*Tehraniet al.* 2015]$^{lc}_{S}$
Spline;	[*Schubertet al.* 2008]$^{*}_{S}$ [*Ziegler&Stiller* 2009]$^{*}_{S}$
Bézier	[*Chenet al.* 2013]$^{lc}_{E}$ [*Huyet al.* 2013]$^{*}_{S}$ [*Choi* 2014]$^{*}_{S}$

Legend: *Use case* – []$_{S}$ = simulation / []$_{E}$ = experimentation; []^{ca} = collision avoidance / []^{cf} = car following / []^{lc} = lane change / []^{m} = merging / []^{o} = overtaking / []^{e} = emergency / []* = no specific use case.

Table 2.3: (continued)

Family	Use case
Mathematical optimization	
LP;	[*Xuet al. 2012*]$^{*}_{E}$ [*Plessenet al. 2017*]$^{o}_{S}$
NLP;	[*Guet al. 2013*]$^{*}_{S}$ [*Kelleret al. 2014*]$^{ca}_{S}$ [*Liet al. 2014*]$^{ca}_{S}$
QP;	[*Zieglaret al. 2014*]$^{*}_{E}$ [*Qianet al. 2016*]$^{o}_{S}$ [*Milleret al. 2018*]$^{*}_{S}$
MPC;	[*Limaet al. 2015a*]$^{*}_{S}$ [*Lefevreet al. 2016*]$^{*}_{E}$ [*Wang&Ayalew 2016*]$^{o}_{S}$ [*Altchet al. 2017*]$^{*}_{S}$ [*Cardosoet al. 2017*]$^{*}_{E}$ [*Liuet al. 2017a*]$^{*}_{S}$ [*Nilssonet al. 2017*]$^{o}_{S}$
DP	[*Choi 2014*]$^{*}_{S}$ [*Guet al. 2016a*]$^{*}_{S}$
AI Logic	
– if-then-rules	[*Vanholmeet al. 2013*]$^{*}_{E}$ [*Constantinet al. 2014*]$^{*}_{S}$ [*Claussmannet al. 2015*]$^{lc}_{E}$ [*Nilssonet al. 2015*]$^{o}_{S}$ [*Nilssonet al. 2016*]$^{lc}_{E}$ [*Liet al. 2017*]$^{*}_{S}$
– FSM	[*Ardeltet al. 2010*]$^{e}_{S}$ [*Zieglaret al. 2014*]$^{*}_{E}$ [*Tehraniet al. 2015*]$^{lc}_{S}$ [*Wanget al. 2015*]$^{*}_{S}$
– Dynamic Bayesian	[*Ulbrich&Maurer 2015*]$^{lc}_{E}$ [*Mouhagiret al. 2016*]$^{o}_{S}$ [*Rodrigueset al. 2016*]$^{*}_{S}$ [*Galceranet al. 2017*]$^{*}_{E}$ [*Zhouet al. 2017*]$^{*}_{S}$ [*Liet al. 2018*]$^{*}_{S}$
AI Heuristic	
– Agent	[*Galceranet al. 2017*]$^{*}_{E}$ [*Liet al. 2018*]$^{*}_{S}$
– SVM	[*Huyet al. 2013*]$^{*}_{S}$ [*Vallonet al. 2017*]$^{lc}_{S}$
– Evolutionary	[*Balujaet al. 1997*]$^{*}_{/}$ [*Onievaet al. 2011*]$^{lc}_{S}$ [*Riazet al. 2015*]$^{ca}_{S}$
AI Approximate	
– Fuzzy logic	[*Naranjoet al. 2008*]$^{o}_{E}$ [*Milanset al. 2011*]$^{*}_{E}$ [*Balalet al. 2016*]$^{lc}_{S}$
– ANN	[*Ngai&Yung 2011*]$^{o}_{S}$ [*Chenet al. 2015b*]$^{*}_{E}$ [*Bojarskiet al. 2016*]$^{*}_{E}$ [*Genget al. 2016*]$^{*}_{S}$ [*Lefevreet al. 2016*]$^{*}_{E}$ [*Rehderet al. 2017*]$^{*}_{E}$ [*Liet al. 2018*]$^{*}_{S}$ [*Wanget al. 2018a*]$^{*}_{S}$ [*Yanget al. 2018*]$^{*}_{E}$
– Belief Network	[*Huanget al. 1994*]$^{*}_{E}$
AI Cognitive	
– Risk	[*Taylor 1964*]$^{*}_{/}$ [*Wilde 1982*]$^{*}_{/}$ [*Ghummanet al. 2008*]$^{ca}_{E}$ [*Prokhorov 2009*]$^{*}_{S}$ [*Ardeltet al. 2010*]$^{e}_{S}$ [*Bautinet al. 2010*]$^{*}_{S}$ [*Glaseret al. 2010*]$^{*}_{S}$ [*Tamkeet al. 2011*]$^{*}_{S}$ [*Danielet al. 2013*]$^{*}_{S}$ [*Wang&Ayalew 2016*]$^{o}_{S}$ [*Naumann&Stiller 2017*]$^{*}_{S}$ [*Wagneret al. 2018*]$^{ca}_{S}$
– Task	[*McKnight&Adams 1970b*]$^{*}_{/}$ [*McKnight&Adams 1970a*]$^{*}_{/}$ [*McKnight&Hundt 1971b*]$^{*}_{/}$ [*McKnight&Hundt 1971a*]$^{*}_{/}$ [*Naranjoet al. 2008*]$^{o}_{E}$ [*Bahramet al. 2014*]$^{*}_{S}$ [*Chenet al. 2015a*]$^{o}_{S}$
– Game	[*Liet al. 2018*]$^{*}_{S}$

Legend: *Use case* – $[]_S$ = simulation / $[]_E$ = experimentation; $[]^{ca}$ = collision avoidance / $[]^{cf}$ = car following / $[]^{lc}$ = lane change / $[]^{m}$ = merging / $[]^{o}$ = overtaking / $[]^{e}$ = emergency / $[]^{*}$ = no specific use case.

Chapter 3

Fuzzy Dempster-Shafer Decision Making

This chapter considers the problem of high-level decision process for autonomous vehicles on highways. The goal is to select a predictive reference trajectory among a set of candidate ones, issued from a trajectory generator. This selection aims at optimizing multi-criteria functions, such as safety, legal rules, preferences and comfort of passengers, or energy consumption. This work introduces a new framework for Multi-Criteria Decision Making to provide a risk assessment. The proposed approach adopts fuzzy logic theory to deal with heterogeneous uncertain criteria and the Dempster-Shafer Theory to aggregate the fuzzy information while considering different sources of risk. Simulation results using existing datasets are presented on car following cases, and extended to lane changing situations.

Contents

The research and development on automated and autonomous vehicles in the last decade may disrupt future transportation utilities. Even if the main focus remains safety on roads to decrease human injuries, Chapter 1 shows that these advanced transportation systems also aim at improving passenger comfort, energy savings, or traffic flow. The addition of those new driving objectives makes it necessary to develop intelligent behaviors with Multi-Criteria Decision Making (MCDM) to autonomous vehicles. The simplified vision of an intelligent system based on Figures 2.2 (p. 16) and 2.3 (p. 17) proposes a 2-level motion planning with a predictive high-level and a reactive low-level. The proposed work focuses on the high-level planning and more specifically on the decision function, through the implementation of MCDM to decide the future motion of the ego vehicle considering the evolution of the surroundings and predefined goals of the ego vehicle.

This chapter is organized as follows. Previous work and motivation are presented in section 3.1. Section 3.2 introduces the multi-criteria selection. The decision algorithm is detailed in section 3.3, and simulation results are discussed in section 3.4. Lastly, section 3.5 presents the conclusion and future work.

3.1 Literature review and motivation

Previous works on the high-level decision making process are reviewed in Chapter 2 with a variety of methods recently applied to autonomous highway driving. These methods cover a large domain of decision mechanisms such as knowledge-based inference engines, heuristic approaches, approximate reasoning, and human-like models. As stated previously, our decision-maker will deal with multi-criteria. The specific issue of MCDM is generally solved in the literature through mathematical optimization [Ali *et al.* 2013, Qian *et al.* 2016] and expert systems [Wardziński 2006, Furda & Vlacic 2010, Chen *et al.* 2014, Noh & An 2018]. The main advantage of these methods is to threat large sets for candidate solutions with a predictive application, meaning a long time horizon as explained in subsection 2.1.2 (p. 15). However, these previous works return criteria evaluations with independently qualified and quantified numerical values. In real experiments, these values are not straightforward to obtain in the case of autonomous vehicles. For example, it is easy to calculate the ego velocity to return a velocity value, but it is harder to associate a value to the dangerousness of the weather conditions.

Moreover, when dealing with vague or imprecise information, linguistic representations of an event may be interpreted using a fuzzy range. This provides the ability to develop a continuous decision model which may have been too complex due to an exhaustive qualification using exact values or too many thresholds boundaries. In that sense, fuzzy logic theory recalls the human evaluation process [Yager & Filev 1994], and proves its relevance for autonomous driving as discussed in [Prokhorov 2009, Balal *et al.* 2016]. Another advantage of fuzzy logic is the consideration of continuous physical variables with a continuous interpretation.

Besides, it is important for the decision-maker to take into consideration how humans interpret the evaluation of a driving situation and what the safety warning means for their own safety. As the human interpretation is typically based on their own cognitive life experiences, a common scale is needed between the human driver and the machine, so that the decision taken by the machine is understandable from the driver point of view. A popular scale in human decision involves risk assessment. Related work on risk estimators in intelligent transportation systems applies with different architectures. The most basic ones use only one risk indicator based on one factor to a specific event [Lefèvre *et al.* 2014]. These approaches are adequate to evaluate the current risk of the situation considering only a rough vehicle collision warning with static or moving surroundings obstacles. In order to provide a more complete risk assessment, estimators could rely on a combination of several factors. Thus, vehicle, driver, and environment data are taken into account to return a suitable risk warning. In this way, the system acquires omniscience and stability based on different sources of risk. For example in [Prokhorov 2009], the author develops a 2-level risk estimation system. The top-level estimator is based on the perceived risk affected by weather, traffic or road conditions, i.e. prior knowledge. The low-level estimator is then applied to refine the top-level estimator with real-time information about the surroundings, such as relative velocities or distances with obstacles. Another 2-level risk estimator proposition in [Daniel *et al.* 2013] calculates a local risk estimator for each category (vehicle, driver, and environment) by separating the static data (e.g. mileage of the vehicle, experience of the driver, road topology), the contextual data (e.g. vigilance of the driver, weather conditions) and the variable data (e.g. accelerations of the vehicle, eye gaze of the driver, Time-To-Collision with obstacles). Each

data is locally fused for each category, defining a local risk. Finally, each category is fused to define a global driving risk, as illustrated in Figure 3.1.

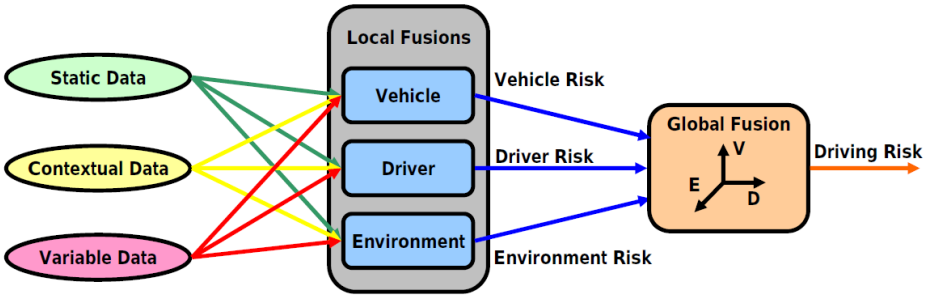


Figure 3.1: The 2-level risk estimation in [Daniel *et al.* 2013].

Furthermore, it is crucial to also consider the uncertainties of criteria values and the consistency of the different sources of risk. Indeed, if a decision evaluation appears to be favorable to a candidate trajectory, but the uncertainties on the criteria values used to evaluate this candidate are strong, the decision might be too risky. In the same way, if the sources for the evaluation depict different conclusions, it may be safer not to make the decision. Thus, uncertainties and consistency impact the risk evaluation of the decision-maker. In order to take these into consideration, the proposed approach incorporates evidential reasoning using the Dempster-Shafer Theory (DST) [Dempster 1967, Shafer 1976]. The DST, or evidential theory or belief theory, is a mathematical framework based on evidences and plausibility in order to combine evidence from different sources and find a degree of belief for the considered event. It has already been adapted for risk assessment on current and predicted driving situation in [Wardziński 2006, Daniel *et al.* 2013].

Our motivation in this work is to develop a high-level MCDM without any restriction on the criteria values, considering both their uncertainties in the form of trustworthiness and strife with a deterministic and straightforward explainable process. To do so, we propose a risk assessment based on the Fuzzy Dempster-Shafer Theory (FDST) approach. Previous works using fuzzy set and evidential theory have been studied in autonomous vehicle decision making, such as the fusion of crisp objective and fuzzy subjective information in [Liu *et al.* 2012], the prediction of road accidents in case of exceeding accelerations in [Gündüz *et al.* 2017], or driving scores for usage based insurance application in [Derbel & Landry 2018]. But, to our knowledge, this work has not been extended to a high-level decision-maker in autonomous vehicles. The presented strategy is based on a 3-level architecture: a set of predefined motions is evaluated with criteria values, which are first fuzzified in local risk levels for each category; second the local risk levels of all the categories are combined as different sources of risk; and third, the motion with the best evaluation is selected as the reference motion over a prediction horizon. The motion can be a set of paths, trajectories or maneuvers, as long as it can be characterized by a set of criteria. The evaluation is based on a linguistic description of the criteria using fuzzy logic and risk assessment combining the belief of risk evidences with FDST. Finally, our algorithm simultaneously considers several factors of risk, their uncertainties, their consistency and their relative importance for human safety in the decision-maker.

3.2 Multi-criteria selection

Multi-criteria involves processing data from different sources, with different units and different range values. In order to process these data correctly, they will first have to be homogenized. Furthermore, not all these data will be of equal importance for the evaluation of the motion and the selection of the decision. This section indicates the preprocessing steps necessary to provide consistent inputs data to our algorithm. Our pretreatment is divided into 4 steps: the data categorization in subsection 3.2.1, the weight assignment in subsection 3.2.2, the predictive characterization in subsection 3.2.3, and the transformation of the crispy criteria values to fuzzy values in subsection 3.2.4.

3.2.1 Criteria categorization

The open environment experienced by autonomous vehicles today is driven by a vast amount of surrounding factors such as vehicles, pedestrians, cyclists, and wildlife. In order to ensure the development of safe autonomous intelligent systems, decisions should be made based on a systemic point of view. That is to say, the decision-maker depends on a global risk assessment over multiple heterogeneous criteria based on the ego and surroundings measurements. For example, [Prokhorov 2009, Daniel *et al.* 2013] take into account the influence of weather, road, traffic, day/time of the date, driver distraction, or demographic data to evaluate the risk of the situation according to three categories: the vehicle, the driver, and the environment. In [Liu *et al.* 2012], the authors prioritize the sensors data as objective information, but agree on the influence of driver subjective data, and separate the risk analysis by defining two categories with objective and subjective data.

Following these reasonings, a linear combination of the risk evaluation of each heterogeneous criterion is not relevant. Thus, we also apply a categorization of the criteria. We choose our classification based on the critical influence of criteria on the future motion. Table 3.1 provides examples of criteria organized in categories. For instance, we distinguish a category for obstacle safety (number 3 in Table 3.1), which gathers the relative velocity, the time headway and the ego velocity, and a category for driving rules (number 4 in Table 3.1), which gathers the legal velocity and legal driving lane on highway environments. In the manuscript, we designate C the total number of the considered categories κ, i.e. $\kappa = 1..C$, and S the total number of the considered criteria s, i.e. $s = 1..S$.

3.2.2 Category and criterion weight

Moreover, as suggested in [Prokhorov 2009], all the criteria do not need to have the same importance for the categories, and all the categories do not present the same local warning risk for the risk assessment. In [Furda & Vlacic 2010], the authors define a hierarchy tree of objectives with attributes assigned to the sub-objectives in the lowest hierarchy level, see Figure 3.2. The attributes of the lowest level are measurable and assigned a weight, which reflects their importance in the overall objective. However, the drawback of such method is to consider only weights on the sub-objectives, identified as the criteria s in our problem, without considering the hierarchy of the objectives, identified as the categories κ in our problem.

Our idea is to consider the relative weight from the point of view of the user's acceptability, i.e how the consequences of this criterion should impact the risk assessment perceived by

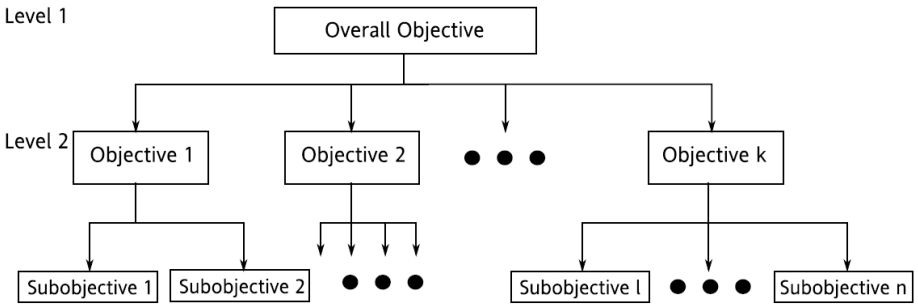


Figure 3.2: The hierarchy of objectives in [Furda & Vlacic 2010].

the driver. In our proposed architecture, the importance of each s criterion (ν_s) is implicitly included using the fuzzy inference system (see subsection 3.3.2). In addition, we define discrete weights w_κ for each κ category, as it can be designed in mathematical optimization approaches, i.e as soft, middle or hard constraints. By definition, the influence of hard constraints cannot be released, whereas it can for soft constraints. Middle constraints influence must be specified considering the severity of the situation. Table 3.1 shows some examples of relative weights with priority to safety. For example, the safety categories are hard whereas the energy savings category is soft. Another advantage of the multi-criteria selection and attribution steps is the ability to relax some criteria in case of a conflicting situation. If the ego vehicle encounters an imminent collision, the energy savings and respecting driving rules do not matter compared to the safety categories.

3.2.3 Predictive criterion

We also consider a predictive approach with candidate trajectories designed over a predicted time horizon (see section 3.4). Criteria values u_s should depict an evaluation of an observable variable over the predicted trajectories. We present in Table 3.1 examples of operators, such as the $maximum/minimum/mean/sum$ value, the end value of the predicted trajectory, or the Boolean $isTrue$ value along the predicted trajectory. For example, for the category 1 vehicle safety, the value for the criterion longitudinal acceleration is calculated as the maximum longitudinal acceleration along the generated motion.

3.2.4 Criterion fuzzy value

Finally, the criteria values u_s issued from the predictive operator are fuzzified on a fuzzy set. This fuzzy set corresponds to the linguistic fuzzy evaluation. In other words, it depicts the fuzzy membership of the criterion value in a fuzzy environment. Even if the fuzzy sets are different for all the criteria, they form an homogeneous interpretation of the criteria in a linguistic and membership manner. For example considering category 4 in Table 3.1, the fuzzy set for legal velocity is being 'too slow', 'slow', 'medium', and 'too fast', whereas the one for legal lane is being 'unsafe' and 'safe'. In fact, the Driver Rules on French highways suggested that the minimum velocity is 60 km/h and the maximum velocity is 130 km/h, and that the vehicles navigates in the most-right lane. If the ego vehicle is below the minimum speed limit it will be 100% 'too slow', if it is at 90 km/h it will be 60% medium and 40%

Table 3.1: Examples of criteria and categories.

Category		**Criterion**		
Weight	**Name**	**Operator**	**Name**	**Fuzzy Set**
Hard	Vehicle safety -1-	*max*	Longitudinal Acceleration/Deceleration	Slow / Medium / High
		max	Lateral Acceleration/Deceleration	Slow / Medium / High
		max	Longitudinal Jerk	Slow / Medium / High
		max	Lateral Jerk	Slow / Medium / High
Hard	Passenger safety -2-	*sum*	Longitudinal Distances	Very Close / Close / Medium / Far / Very Far
		min	Lateral Distances	Very Close / Close / Medium / Far / Very Far
Hard	Obstacle safety -3-	*mean*	Relative Velocity	Negative / Slow / Medium / Fast
		min	Time Headway	Unsafe / Safe / Safer / Very safe
		end	Ego Velocity	Slow / Medium / Fast
Middle	Driving rules -4-	*min&max*	Legal Velocity	Too Slow / Slow / Medium / Too Fast
		isTrue	Legal Lane	Unsafe / Safe
Soft	Reference wishes -5-	*end*	Direction	Over-right / Right / Middle / Left / Over-left
		end	Time	Too Long / Long / Medium / Quicker
		mean	Velocity	Too Slow / Slow / Medium / Too Fast
		mean	Acceleration/Deceleration	Slow / Medium / High
		min	Distances	Very Close / Close / Medium / Far / Very Far
Soft	Passenger comfort -6-	*max*	Longitudinal Acceleration/Deceleration	Slow / Medium / High
		max	Lateral Acceleration/Deceleration	Slow / Medium / High
		min	Distances	Very Close / Close / Medium / Far / Very Far
		mean	Road Position	Over-right / Right / Good / Left / Over-left
Soft	Navigation indication -7-	*end*	Direction	Over-right / Right / Good / Left / Over-left
Soft	Energy savings -8-	*mean*	Longitudinal Acceleration/Deceleration	Slow / Medium / High

slow for instance; if it navigates in the mostright lane, it will be 100% safe, in the middle lane, 50% safe and 50% unsafe.

The predictive criteria and their fuzzy sets are the inputs of the Fuzzy Dempster-Shafer algorithm presented in the next section.

3.3 Fuzzy Dempster-Shafer algorithm

The decision-maker is based on the previously mentioned criteria $s = 1..S$ in section 3.2. They represent the driving scene with data issued from exteroceptive (environment information) and proprioceptive (ego information) sensors. Thus, the decision process is based on imperfect and heterogeneous information provided by more or less reliable and conflicting sources. The aim of our algorithm is to combine different criteria values u_s with their uncertainties to yield a risk assessment RA. To do so, we develop the architecture depicted in Figure 3.3 used for each candidate trajectory $p = 1..P$. We first define in level 1 a Fuzzy Inference System (FIS) on the criteria $s = 1..S$ to return fuzzy local warning risk levels LW_κ^ι for each category $\kappa = 1..C$ with a common scale of output fuzzy sets ι. Level 2 applies the Fuzzy Dempster-Shafer Theory (FDST) to combine the resulting risk levels LW_κ^ι, and calculates the belief and plausibility degree $RA[\text{bel}, \text{pl}]_p^h$ of each predefined risk hypotheses $\mathcal{H}_{h=1..H}$. Lastly, level 3 evaluates a risk indicator $\mathcal{I}_p$ to determine the best trajectory. This section will first recall the basic concept of the two theories of fuzzy logic and evidential reasoning in subsection 3.3.1, and then develop each of the 3-level FDST architecture (level 1 in subsection 3.3.2, level 2 in subsection 3.3.3, and level 3 in subsection 3.3.4).

3.3.1 Basic concepts

FDST is based on the combination of 2 theories: fuzzy logic and evidential reasoning. The idea is to employ the Dempster-Shafer Theory (DST) to aggregate the LW_κ^ι for all the categories κ to return the risk assessment RA. This combination has three main advantages: (1) it is deterministic and explainable as the decision results from a rule-base; (2) it avoids the defuzzification step in the fuzzy logic process, and thus retains the principle of fuzzy set; and (3) the output of evidential reasoning matches with the vague interpretation of fuzzy logic. Indeed, the input data received by the system belongs to many conclusions at the same time. For example, a velocity of 110 km/h is both medium and high, whereas a velocity of 90 km/h is both medium and small. Thus there exist a more or less degree of membership to each conclusion, which can be interpreted as a plausibility for each conclusion. Combining the different conclusions of each input data results in an interval of plausibility for the conclusions. This corresponds to the output process of DST. This subsection summarizes the underlying concepts of the Fuzzy Logic (3.3.1.1) and the Dempster-Shafer Theory (3.3.1.2).

3.3.1.1 Fuzzy Logic

The use of fuzzy set theory for decision making was first introduced by L. A. Zadeh in [Zadeh 1965], and has become a very popular method in system control. This type of decision making can be used to represent complex problems involving human reasoning which can be difficult using classic models [Yager & Filev 1994].

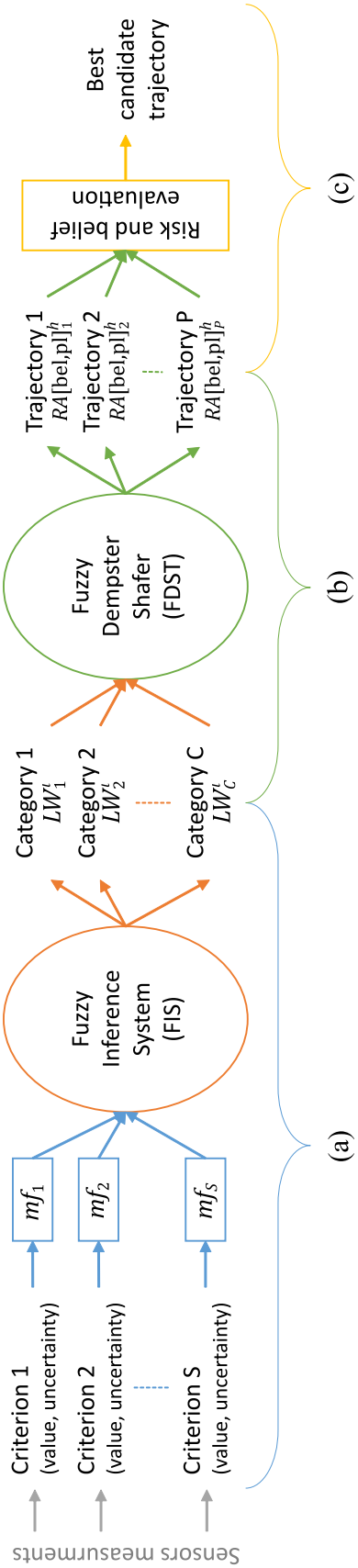


Figure 3.3: FDST algorithm architecture. (a) Level 1: fuzzification with the membership functions μ and membership degrees mf of the S criteria values $u_{s=1..S}$ and uncertainties in order to return for each category $\kappa = 1..C$ the ι local warning risk levels LW_κ^ι. (b) Level 2: application of FDST on the LW_κ^ι to return for each trajectory $p = 1..P$ the risk assessment RA for each risk hypothesis $h = 1..H$ with the decision brackets $RA[\text{bel, pl}]_p^h$. (c) Level 3: calculation of the best candidate trajectory based on $RA[\text{bel, pl}]_{p=1..P}^{h=1..H}$.

As an overview, fuzzy logic provides the ability to represent vague and imprecise variables in a decision making system. Unlike the traditional crisp sets, which have strict boundaries, i.e. the variable belongs or belongs not to a set so that the membership is binary, fuzzy logic defines a continuous membership degree mf to a fuzzy set described by a membership function μ for each fuzzy set. The fuzzy sets can have overlapping boundaries, which means that the variable can have more than one non-zero membership degree. The input space is then mapped to the output space using a list of if-then statements called rules. Hence, the rules do not refer to thresholds but to the corresponding mf from input to output spaces. Various methods have been used in research to design and optimize μ and determine the systems' rule-base including expert knowledge [Yager & Filev 1994], clustering [Yager & Fileu 1993], and adaptive neuro-fuzzy inference systems [Jang 1993]. Different types of inference systems also exist and vary in the way outputs are determined. The most popular ones are the *Mamdani* and the *Sugeno* FIS [Yager & Filev 1994]. The main difference between them is the output membership function μ_{out}. The Sugeno output membership functions are either linear or constant, whereas the Mamdani ones are fuzzy sets. In this proposed framework, we want to fuzzify, homogenize, and combine the chosen criteria values $s = 1..S$ into categories $\kappa = 1..C$, see Figure 3.3a. We define the output μ_{out} to return local warning risk levels (LW^{ι}_{κ}) as fuzzy sets, thus a *Mamdani-type* inference is developed. In the rest of the manuscript, μ and mf designate respectively the membership functions and the membership degrees on the entire fuzzy set; $\mu_A(s)$ and $mf_A(u_s)$ are the membership function and the membership degree of the fuzzy set A for the input criterion s with the criterion value u_s.

The development of a Mamdani Fuzzy Inference System (FIS) generally includes the following five steps ([Yager & Filev 1994]):

- (i) input and output membership functions (μ) design,
- (ii) fuzzification,
- (iii) rule-base development,
- (iv) rules aggregation,
- (v) output defuzzification.

First (i), each input and output are expressed in fuzzy sets. Each fuzzy set is represented by a membership function μ over a universe of discernment uod. The universe of discernment uod is the range of the variable to analyze. Each input and output has a corresponding uod. A membership function μ is an arbitrary curve that maps each element of the uod to a membership value between 0 and 1, called the membership degree mf. Thus, a fuzzy set A is represented by its membership function μ_A for all the elements z in its uod_A in Equation 3.1:

$$A = \{(x, \mu_A(x)) | x \in uod_A\} \tag{3.1}$$

The second step (ii) of fuzzification consists in associating the input variable value to its input membership degrees. The membership degrees mf are the intersections between the membership functions μ and the input variable value u_s in the uod. Due to the overlapping boundaries of μ, the fuzzified input value can belong to more than one μ, and thus may have more than one membership degree mf for each of the corresponding input fuzzy sets.

Third (iii), a rule-base describes the relationship between the input fuzzy sets and the output fuzzy sets. This rule-base is expressed as if-then statements with the logic operator AND, OR, NOT. The rules express the relationship between the input fuzzy sets of each input variable and the output fuzzy sets. The activation of a rule depends on the membership degree $mf(u_s)$ of its input variables u_s previously calculated in the second step (ii). Here is an example of a FIS with 2 inputs (2 criteria $s = 1, 2$) and 1 output (RA); input 1 is fuzzified with 3 fuzzy sets (small, medium, high), input 2 is fuzzified with 2 fuzzy sets (low, high), and output is fuzzified with 4 fuzzy sets (very bad, bad, good, very good). The rule number 1 "*If* Input 1 is small AND Input 2 is low, *then* output RA is good" is activated if $(mf_{1(small)}(u_1) \neq 0$ & $mf_{2(low)}(u_2) \neq 0)$. In the manuscript, we named IRR the set of the activated rules and $IRR_r \equiv rule_r(mf_s(u_s))$ the activation of rule number r with the corresponding membership degree $mf_s(u_s)$ for each input criterion s activating the rule.

After the set of activated rules IRR have been deduced from the membership degree mf of the input fuzzy sets and the rule-base, the rule-implication calculates the mf of the fuzzy output of each activated rule. It is decomposed in two steps. The first step applies the logic operator of the rules on the membership degrees of the input values. A classic interpretation of the logic operator AND, OR, NOT are respectively the minimum, maximum and additive complement functions. With the previous example, the resulting value of the rule "*If* Input 1 is small AND Input 2 is low, *then* output RA is good" is equal to $min(mf_{1(small)}(u_1), mf_{2(low)}(u_2))$. The second step is the truncation of the corresponding output fuzzy set with this resulting value. In our previous example, the membership function $\mu_{out(good)}$ of the output fuzzy set *good* is truncated by the $min(mf_{1(small)}(u_1), mf_{2(low)}(u_2))$ value. In the manuscript, we name ORR the set of the implicated rules and $ORR_r \equiv trunc(\mu_{out}, \mathcal{T}(IRR_r))$ the truncated output fuzzy set corresponding to rule number r with $\mathcal{T}$ its logic operator.

The fourth step (iv) aggregates the truncated output fuzzy sets of all the activated rules ORR, calculated in the implication process. In the manuscript, we named ARR the aggregation function of the implicated rules. There exist different aggregation operator such as the maximum, the probabilistic or, or the sum of each activated rule's output fuzzy set. With $\mathcal{S}$ the aggregation operator, and $r1$, $r2$ the two activated rules, $ARR = \mathcal{S}(ORR_{r1}, ORR_{r2})$.

The final step (v), defuzzification, provides a single output value from ARR; however, the proposed architecture uses the aggregation function ARR for the FDST algorithm. Thereby, each category κ is assigned with a LW_κ^ι as the likelihood probability of being in each predefined risk level ι as defined in the rule-base.

3.3.1.2 Evidential reasoning

The belief theory is based on the combination of evidences. It uses the definition and comparison of belief functions in order to come to a plausible reasoning [Shafer 1976]. In this part, we recall the vocabulary of the Dempster-Schafer Theory (DST).

In DST, the set of evidences must be defined on the same universe X, called the frame of discernment. Under closed-word assumption, the frame of discernment is thus composed of all the possible solutions. We call $P(X)$ (or 2^X) the power set, which represents the set of subsets of X plus the empty set $\varnothing$. Each subset A of $P(X)$ is assigned a mass m, which corresponds to a degree of belief, represented by a belief function or the basic belief

assignment *bba*. The mass of a subset element represents the proportion of the other subsets asserting that the current state is part of the considered element and no other subset, i.e. the veracity of a proposition. The nonzero mass sets are called *focal elements*. The mathematical formalism is then defined as:

$$m : 2^X \rightarrow [0,1]; m(\varnothing) = 0; \sum_{A \subseteq 2^X} m(A) = 1 \tag{3.2}$$

The interest of belief theory is to combine evidences to find their common part within the universe. The Dempster's combination rule of evidence for independent sources 1 and 2 is given in [Dempster 1967] by the joint mass $m_{1,2}$:

$$\begin{aligned} m_{1,2}(\varnothing) &= 0 \\ m_{1,2}(A) &= (m_1 \oplus m_2)(A) \\ &= \frac{1}{1-K} \sum_{B \cap C = A \neq \varnothing} m_1(B) m_2(C) \end{aligned} \tag{3.3}$$

where K represents a measure of conflict between the two mass sets: $K = \sum_{B \cap C = \varnothing} m_1(B) m_2(C)$, with B and C respectively the subsets of sources 1 and 2. In practice, the independence of sources is not always verified, and conflicts have different interpretation. This can be handled by extending the original combination rules. A non-exhaustive list of rules for the combination of evidences is presented in [Sentz *et al.* 2002].

The use of DST differs from the classic probabilistic theory, in the sense that the exact probability P of an event A is contained in an interval, bounded by a lower probability called belief (bel) and an upper probability called plausibility (pl):

$$\text{bel}(A) \leq P(A) \leq \text{pl}(A) \tag{3.4}$$

The belief represents the sum of the masses of all the subsets of the considered event, i.e. the sum of the masses of all the focal elements which necessarily imply the desired event. The plausibility represents the sum of the masses of the sets that intersect the considered event, i.e. the sum of the masses of all the focal elements that do not necessarily contradict the desired event. In other words:

$$\begin{aligned} \text{bel}(A) &= \sum_{B | B \subseteq A} m(B) \\ \text{pl}(A) &= \sum_{B | B \cap A \neq \varnothing} m(B) \end{aligned} \tag{3.5}$$

In our work, the evidences describe driving scene data, *i.e* the previously defined categories $\kappa = 1..C$ in section 3.2. Each element of the universe is interpreted as a local warning risk level (LW_κ^ι), which ranges over the interval $[0,1]$, 0 corresponding to a zero risk level and 1 being the highest risk level. The mass is calculated with the output membership function ARR of the fuzzy logic. More details and examples are given in subsection 3.3.3.

3.3.2 Level 1: Fuzzy Inference System

The first level corresponds to the fuzzification step to read and transpose heterogeneous criteria $s = 1..S$ into homogeneous categories $\kappa = 1..C$, see Figure 3.3a. We follow the steps 1 to 4 from the fuzzy logic process described in subsection 3.3.1.1. A specific example is given here with category 4 – Driving rules in Table 3.1.

For the first step, input and output membership function μ design, we assume that all the input and output μ are provided for each criterion s and each local warning risk level (LW_κ^ι). The heterogeneousness of the criteria $s = 1..S$ results from the different fuzzy sets, μ and universe of discernment uod for each criterion s. On the contrary, the homogeneousness of the categories $\kappa = 1..C$ leads to the same uod, μ and fuzzy sets ι to define the LW_κ^ι for all the categories C.

In our work, we define 5 μ_{out} outputs warning levels to estimate the risk over $uod_{out} = [0, 1]$: very low ($\mu_{out(veryLow)}$, $uod_{veryLow} = [0, 0.25]$), low ($\mu_{out(low)}$, $uod_{low} = [0, 0.5]$), medium ($\mu_{out(med)}$, $uod_{med} = [0.25, 0.75]$), high ($\mu_{out(high)}$, $uod_{high} = [0.5, 1]$) and very high ($\mu_{out(veryHigh)}$, $uod_{veryHigh} = [0.75, 1]$), as shown in Figure 3.4. In the example of category 4 – Driving rules in Table 3.1, the first criterion *Legal velocity (Lv)* has a $uod_1 = [0, 180]$km/h and four μ ($\mu_{1(tooslowVel)}$, $\mu_{1(slowVel)}$, $\mu_{1(medVel)}$, $\mu_{1(toofastVel)}$), shown in Figure 3.5a, and the second criterion *Legal lane (Ll)* has a $uod_2 = [0, 1]$ and two μ ($\mu_{2(unsafeLane)}$, $\mu_{2(safeLane)}$), shown in Figure 3.5b.

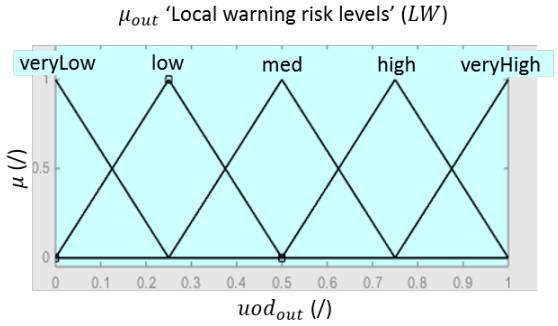
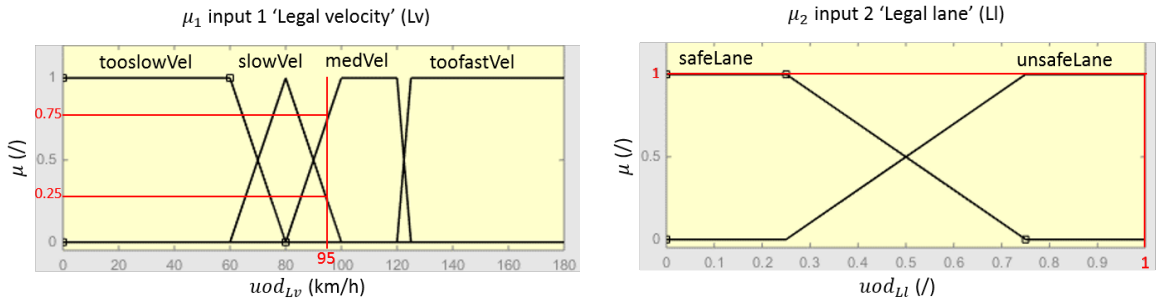


Figure 3.4: Fuzzy outputs μ for the warning levels LW_κ^ι such that $\forall\kappa \in [1, C], \iota \in \{veryLow, low, med, high, veryHigh\}$ in $uod = [0, 1]$.

The second step of fuzzification consists on defining the membership of each considered criterion. For example, the category 4 – Driving rules has two criteria: *Legal velocity (Lv)* and *Legal lane (Ll)*. If the ego vehicle has a velocity of 95km/h and is on the mostleft lane, the membership degrees to the input fuzzy sets are equal to $mf_{1(slow)} = 0.25$ and $mf_{1(med)} = 0.75$ for the criterion 1 *Lv* and $mf_{2(unsafe)} = 1$ for the criterion 2 *Ll*, with respect to the input membership functions defining in Figures 3.5a and 3.5b.

Furthermore, we are interested in uncertainties based on a trust value to quantify the fidelity of the measurement of an input variable, i.e. to attribute a confidence degree to the variable. This trust value lies between 0 and 1. The closer to 1 the trust value is, the higher the confidence degree of the measurement is. Thus, the degree of membership of each fuzzy set for an input variable s after the input fuzzification step is weighted by the the importance



(a) Four μ for criterion 1 legal velocity: too slow ($\mu_{1(tooslowVel)}$), slow ($\mu_{1(slowVel)}$), medium ($\mu_{1(medVel)}$) and too fast ($\mu_{1(toofastVel)}$).

(b) Two μ for criterion 2 legal lane: safe ($\mu_{2(safeLane)}$) and unsafe ($\mu_{2(unsafeLane)}$).

Figure 3.5: Membership functions for category 4 – Driving rules.

of each criterion ν_s, which is the trust value in our problem, as stated in Equation 3.6:

$$\begin{cases} \mu'_s = \nu_s * \mu_s, \\ mf'_s = \nu_s * mf_s, \end{cases} \tag{3.6}$$

where μ and mf are the membership function and membership degree before weighting, μ' and mf' the membership function and membership degree after weighting.

In the third step, the rule-base links the input membership functions μ_s of criteria $s = 1..S$ to the output ones μ_{out}. For example, we assign the rule-base in Table 3.2 for category 4 – Driving rules.

Table 3.2: AND rule-base development for category 4 – Driving rules.

N. rule	Input 1: Lv (km/h)	Logic operator	Input 2: Ll (/)	Output: LW_4
1	tooslowVel	AND	/	veryHigh
2	toofastVel	AND	/	veryHigh
3	slowVel	AND	safeLane	med
4	**slowVel**	AND	**unsafeLane**	**high**
5	medVel	AND	safeLane	low
6	**medVel**	AND	**unsafeLane**	**veryLow**

Considering the previous example with $Lv = 95$ km/h and $Ll = 1$, the rules number 4 and 6 are activated with the inputs $mf_1(Lv = 95)$ and $mf_2(Ll = 1)$ in Equation 3.7:

$$IRR = IRR_4 \ \& \ IRR_6 = \text{rule}_4(0.25, 1) \ \& \ \text{rule}_6(0.75, 1) \tag{3.7}$$

The result of evaluating the rules r through the implication method AND (equivalent to $\mathcal{T} \equiv$ min function) on IRR_r are the truncated output μ_{out} for ORR_r expressed in Equa-

tion 3.8:

$$ORR = ORR_4 \ \& \ ORR_6, \tag{3.8}$$

$$with \begin{cases} ORR_4 = trunc(\mu_{\text{high}}, \min(0.25, 1)), \\ ORR_6 = trunc(\mu_{\text{veryLow}}, \min(0.75, 1)), \end{cases} \tag{3.9}$$

with $trunc(g, a)$ the truncated function g by the threshold value a.

The result of the fourth step, evaluating the truncated output membership functions ORR_r for each activated rule r through the aggregation method $\mathcal{S} \equiv \max$ for ARR, is the maximum value of the ORR over the output uod expressed by Equation 3.10:

$$ARR = \max(ORR_4, ORR_6) \tag{3.10}$$

The corresponding figures of IRR, ORR and ARR calculation are shown in Figure 3.6:

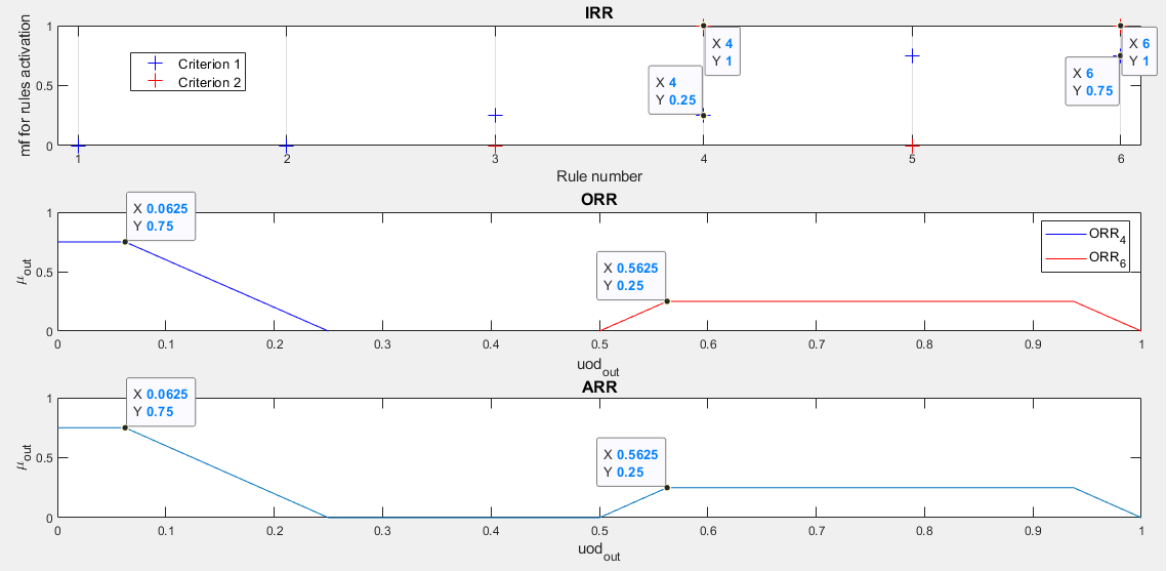


Figure 3.6: Fuzzy steps 3 and 4 for category 4 – Driving rules.

Finally, the aggregation output for the local warning risk levels of category 4 is represented in Figure 3.7 by overlapping the ARR function and the output fuzzy sets in Figure 3.4. The blue area corresponds to $LW_4^{veryLow}$, the pink area to LW_4^{low}, the yellow area to LW_4^{med}, the green area to LW_4^{high} and the purple area to $LW_4^{veryHigh}$ for the input values $Lv = 95$ km/h and $Ll = 1$.

The covered areas for each of the LW_κ^ι are then used as the inputs for the level 2 of our architecture, explained in details in the next subsection 3.3.3.

3.3.3 Level 2: Risk Assessment

The second level of our architecture returns a risk assessment RA for each trajectory $p = 1..P$, see Figure 3.3c (p. 57).The aim of risk assessment using the theory of evidence is to characterize the similarity of events obtained by the combination of the *bba* from evidences within the

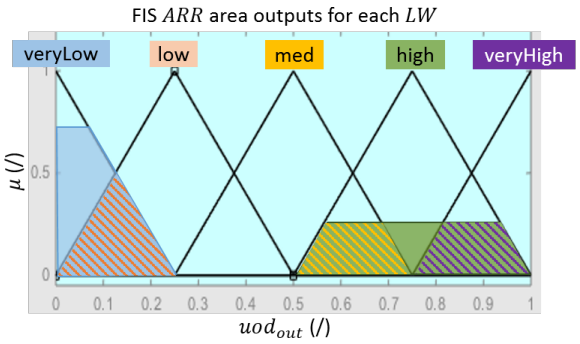


Figure 3.7: FIS ARR area outputs for each LW for category 4 – Driving rules.

risk hypotheses $\mathcal{H}_{h=1..H}$. The procedure is: (i) defining H hypotheses, (ii) constructing the focal elements of evidences and their *bba*, (iii) combining *bba*, and (iv) calculating the resulting risk assessment RA for each hypothesis h of each trajectory p. In our case, the events are the different local warning risk levels (LW_κ^ι) issued from the focal elements of the evidences, the evidences are the categories $\kappa = 1..C$ defined in section 3.2, and the hypotheses $\mathcal{H}_{h=1..H}$ are a priori defined as the values for the risk assessment (RA) acceptable to non-acceptable, characterized by the evidential decision brackets $RA[\text{bel}_p^h, \text{pl}_p^h]$, see Figure 3.3b.

3.3.3.1 Hypotheses

In this work, belief theory is used to define a risk assessment RA for each candidate trajectory p. One of the drawbacks of risk assessment is the need for defining an acceptable or unacceptable risk threshold, which depends heavily on the driver's experience, as analyzed in [Wardziński 2006]. The use of belief theory with fuzzy sets makes it possible to handle fuzzy thresholds too. Thus, the advantage of the proposed architecture is to adjust the thresholds' shapes to a continuous risk tolerance.

Considering the three threat levels in [Noh & An 2018], we represent 3 hypotheses $\mathcal{H}_{h=1..3}$ for an acceptable level of risk (A), a partially acceptable (PA) and a non-acceptable (NA), i.e. $\mathcal{H} = \{A, PA, NA\}$. We consider the respective trapezoidal functions f_h depicted in Figure 3.8, based on the local warning risk levels LW_κ^ι defined in the previous level 1 in subsection 3.3.2. Indeed, we define as acceptable a very low risk level, 5/6 of a low risk level and a sixth of medium one, the partially acceptable hypothesis covers the last sixth of low, medium and the first half of high, and the non acceptable hypothesis is given by the second part of high and very high risk levels. This choice has been made considering a conservative driver profile and the safety rules of the Driver Rules. Therefore, the transition between A and PA is smooth due to the driver profile, whereas the transition PA to NA is radical due to the safety approach.

3.3.3.2 Evidence and Belief functions

The first key point of evidential reasoning concerns the basic belief assignment (*bba*) on the focal elements of evidences. In our application, the focal elements of an evidence are the outputs $\iota \in \{veryLow, low, med, high, veryHigh\}$ local warning risk levels (LW_κ^ι) of a

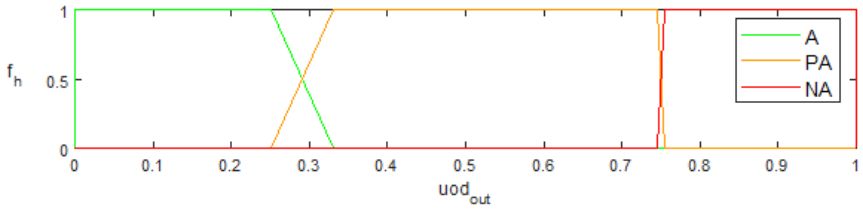


Figure 3.8: Representation of the 3 hypotheses functions (A, PA, NA).

category κ. The LW_κ^ι result from the output μ and mf calculated in subsection 3.3.2. For each category κ, several local warning risk levels may then be concerned.

It was decided here to consider, a priori, all LW_κ^ι as having equal probability for each category κ. However, each category κ is also assessed from the fuzzy combination of criteria s. The aggregate output of fuzzy rules is therefore interpreted as equivalent to the relative probability that each local warning risk level LW_κ^ι of the category κ is represented. Thus, the new mass of each warning level ι of each category κ is the product of the uniform distribution and the surface covered by the ARR function (i.e. the integral value) on the corresponding LW_κ^ι [Jiang *et al.* 2012]. These new masses are finally normalized in order to respect Equation 3.2 (subsection 3.3.1.2, p. 60). The *bba* calculation for a warning level ι for a category κ over uod_{out} is summarized in Equation 3.11:

$$bba_\kappa^\iota = \frac{1}{n_{LW}} \int ARR(LW_\kappa^\iota)\,\mathrm{d}(uod_{out}), \tag{3.11}$$

with n_{LW} the cardinal of the output fuzzy sets, i.e. the number of local warning risk levels.

In the example of category 4 – Driving rules, the *bba* shown in Figure 3.4 are calculated in Equation 3.12:

$$\begin{cases} bba_4^{veryLow} = \frac{1}{5} * \int ARR(LW_4^{veryLow})\,\mathrm{d}(uod_{out}) = 0.3061, \\ bba_4^{low} = \frac{1}{5} * \int ARR(LW_4^{low})\,\mathrm{d}(uod_{out}) = 0.1632, \\ bba_4^{med} = \frac{1}{5} * \int ARR(LW_4^{med})\,\mathrm{d}(uod_{out}) = 0.1225, \\ bba_4^{high} = \frac{1}{5} * \int ARR(LW_4^{high})\,\mathrm{d}(uod_{out}) = 0.2857, \\ bba_4^{veryHigh} = \frac{1}{5} * \int ARR(LW_4^{veryHigh})\,\mathrm{d}(uod_{out}) = 0.1225. \end{cases} \tag{3.12}$$

3.3.3.3 Combining Beliefs

The second key point is the combination of evidences. In our case, we combine the LW_κ^ι to determine the beliefs of the risk assessment $RA[bel, pl]_p^h$ for each hypothesis $\mathcal{H}_{h=A,PA,NA}$ for each trajectory $p = 1..P$.

The combination scheme is the same as the one depicted in [Gündüz *et al.* 2017], i.e. the joint masses are recursively computed between the focal elements two by two. We calculate all the possible combinations of each focal element in each evidence with the other

focal elements of all other evidences. In our problem, this combination approach consists in calculating the resulting membership functions μ and basic belief assignment bba of each non-zero local warning risk levels LW_{κ}^{ι} from each category κ with the non-zero $LW_{\kappa'}^{\iota}$ of all the other categories.

Besides, in our application, the conflict situation leads to inconsistent decisions not to be ignored. Indeed, if a first category κ has only one focal element describing a very high risk level (i.e. only $LW_{\kappa}^{veryHigh} \neq 0$), and a second category κ' has its only focal element describing a very low risk (i.e. only $LW_{\kappa'}^{veryLow} \neq 0$), it means that the global estimated risk is both very high and very low. In this situation, the 2 evidences' conclusions are in conflict; this conflict leads to an incoherent global risk level decision making. In order to address this issue of conflict, we apply the conjunctive Yager's combination rule [Sentz *et al.* 2002] in Equation 3.13. The difference here is to not normalize the joint mass with the Dempster's conflict factor K in Equation 3.3, but to translate conflict in ignorance. As a consequence, the value of bel is smaller and the value of pl larger than the one obtained with the Dempster's rule.

Moreover, we introduce in Yager rule's combination the weight of categories w_{κ} in the combination of μ, as it has been suggested in [Liu *et al.* 2012]. The idea is to shift the resultant membership function in the direction of the highest-weight category. Consequently, the LW of the highest-weight category support a higher contribution on the calculation of the resulting beliefs.

Finally, the combination of a category 1 with the focal element LW_1^{ι} and a category 2 with the focal element $LW_2^{\iota'}$ results in $\mu_{1,2(\iota,\iota')}$ over $uod_{1,2(\iota,\iota')} = [0,1]$ and $bba_{1,2(\iota,\iota')}$ given in Equation 3.13:

$$\begin{cases} \mu_{1,2(\iota,\iota')} = G \times \min\left(w_1\mu_{1(\iota)}, w_2\mu_{2(\iota')}\right) \\ bba_{1,2(\iota,\iota')} = bba_{1(\iota)} \times bba_{2(\iota')} \end{cases} \tag{3.13}$$

where G is a normalization factor defined in Equation 3.14, such that the category weights w_{κ} only influence the μ representation and not the bba:

$$G = \frac{\max(\min(\mu_{1(\iota)}, \mu_{2(\iota')}))}{\max(\min(w_1\mu_{1(\iota)}, w_2\mu_{2(\iota')}))}. \tag{3.14}$$

Figure 3.9 illustrates the calculation of the resulting membership function μ for 2 categories $\kappa = 1, 2$, with $w_1 = w_2 = 1$.

Furthermore, in our example, the combination operator min for $\mu_{\kappa,\kappa'}$ with the intersection of the focal elements is symmetric thanks to the identical output fuzzy sets LW_{κ}^{ι} definition for all the categories $\kappa = 1..C$, as illustrated in Figure 3.10. Moreover, the uod_{out} for each $\iota \in \{veryLow, low, med, high, veryHigh\}$ do not overlap for all the combination, as depicted in Figure 3.10. As a consequence of the min combination operator, the red combinations in the matrix of Figure 3.10 are equal to 0. Indeed, the min resulting membership function between $\mu_{out,low}$ over $uod_{low} = [0, 0.5]$ and $\mu_{out,high}$ over $uod_{high} = [0.5, 1]$ is equal to 0. Therefore, for any combination of C categories, the combined $\mu_{\kappa,\kappa'}$ are calculated for the following fuzzy sets: $\{veryLow, veryLow/low, low, low/med, med, med/high, high, high/veryHigh, veryHigh\}$. Those fuzzy sets contribute to the belief-plausibility interval of each hypotheses $\mathcal{H}_h$, as detailed in the next subsection 3.3.3.4.

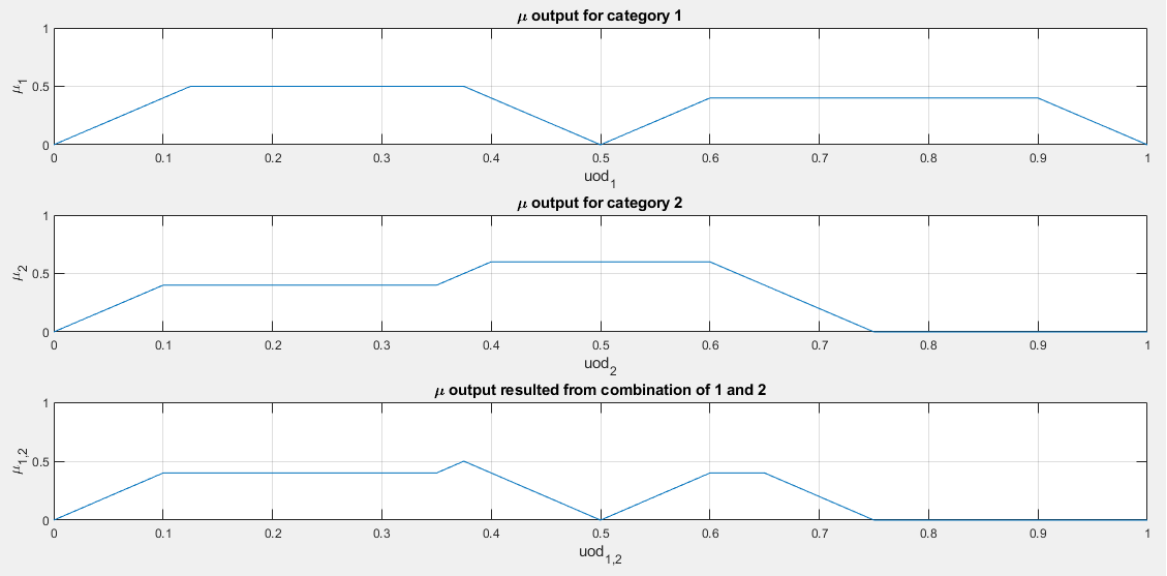


Figure 3.9: Example of the μ combination process.

3.3.3.4 Risk evaluation

The DST returns the belief-plausibility interval [bel, pl] containing the exact probability P of the state A. Two pieces of information must be interpreted: (i) the smaller the interval is, the stronger the knowledge on the probability of being in this state is, (ii) the closer the interval is to 1, the greater the probability of being in the state is. The value of bel and pl can then be interpreted as respectively the conservative and the risky parts of the decision.

The fuzzy sets which contributes to the calculation of the the belief-plausibility intervals of the hypotheses $\mathcal{H}_{h\in\{A,PA,NA\}}$ in Figure 3.8 are issued from the combination of the categories detailed in the previous subsection 3.3.3.3, which return $n_{LW} = 9$ joint focal elements $\{veryLow, veryLow/low, low, low/med, med, med/high, high, high/veryHigh, veryHigh\}$.

In order to respect Equation 3.2 (subsection 3.3.1.2, p. 60), the joint focal elements $l = 1..n_{LW}$ are first normalized in Equation 3.15:

$$\begin{cases} \mu'_l = \dfrac{\mu_l}{\max(\mu_l)} \\ bba'_l = \max(\mu_l) \times bba_l \end{cases} \tag{3.15}$$

For the sake of clarity, we displayed separately the contribution $\{veryLow, low, med, high, veryHigh\}$ in Figure 3.11 and $\{veryLow/low, low/med, med/high, high/veryHigh\}$ in Figure 3.12. The resulting μ_{out} of the example shown in Figure 3.9 is represented in a solid blue line in the top representation of both figures. The contributions of $\{veryLow, low, med, high, veryHigh\}$ and $\{veryLow/low, low/med, med/high, high/veryHigh\}$ are plotted respectively in colored dashed lines in the center representations in Figures 3.11 and 3.12. Each contributed $mf_{out,l}$ is normalized, as shown in the colored solid lines in the center representations in Figures 3.11 and 3.12. The bottom representations report the contribution of the normalized mf'_{out} to the output fuzzy sets $\{veryLow, low, med, high, veryHigh\}$ and $\{veryLow/low, low/med, med/high, high/veryHigh\}$.

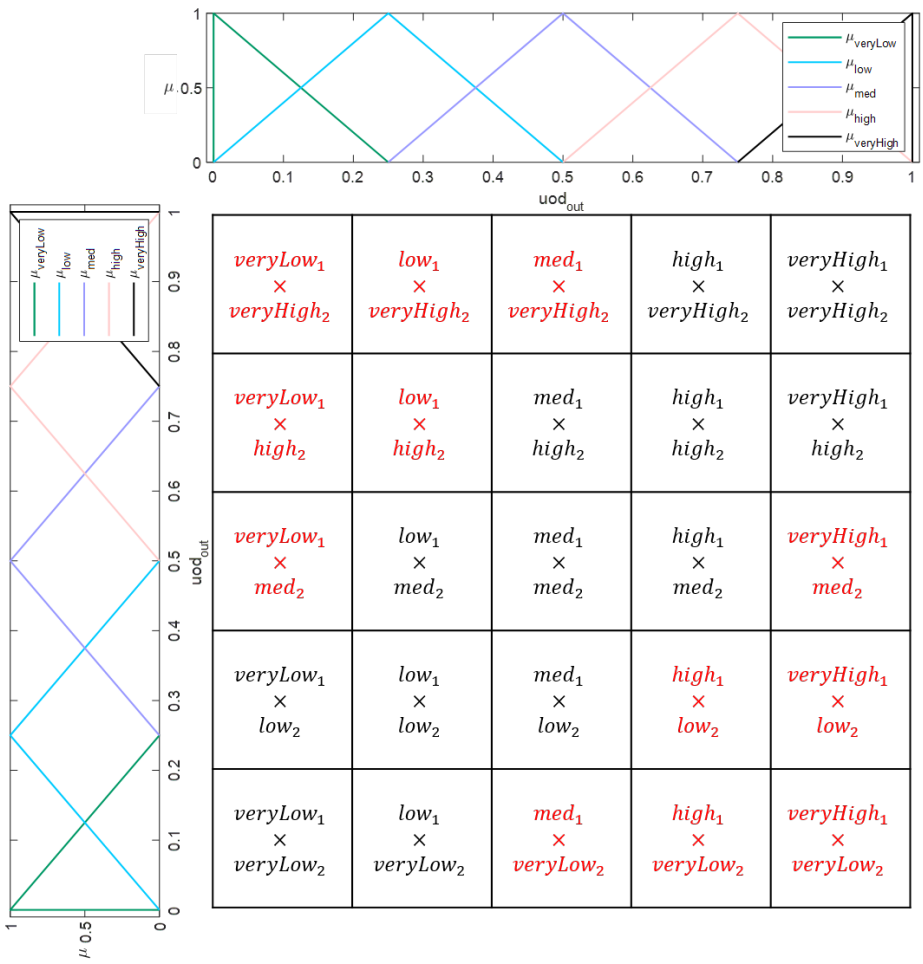


Figure 3.10: Illustration of the μ combination process. The red combinations are equal to 0, as their *uod* do not intersect.

To calculate the contribution of fuzzy sets to the hypotheses, we use the generalization of the Dempster-Shafer Theory to fuzzy sets in [Yen 1990]. It consists in the decomposition of a fuzzy focal element A as a collection of nonfuzzy α-level subsets. Even if the hypotheses and the fuzzy sets are represented by simple trapezoidal and triangular shapes in our problem, we use the α-level subsets for numerical approximation to calculate the surface of arbitrary functions. An illustration of the contribution of $\{veryLow, low, med, high, veryHigh\}$ to the three hypotheses $\mathcal{H}_{h\in\{A,PA,NA\}}$ and 5 α-level subsets is shown in Figure 3.13.

Finally, the belief and plausibility values for evaluating the risk assessment $RA[\text{bel}, \text{pl}]_p^h$ for each hypothesis $\mathcal{H}_{h\in A,PA,NA}$ of each trajectory $p = 1..P$ are obtained in Equation 3.16:

$$\begin{aligned} \text{bel}\left(\mathcal{H}_h\right) &= \sum_{l=1}^{n_{LW}} bba_l' \times \left(\sum_{\alpha_i} (\alpha_i - \alpha_{i-1}) \times \inf_{z|\mu_l'(z)>\alpha_i} f_h(z) \right) \\ \text{pl}\left(\mathcal{H}_h\right) &= \sum_{l=1}^{n_{LW}} bba_l' \times \left(\sum_{\alpha_i} (\alpha_i - \alpha_{i-1}) \times \sup_{z|\mu_l'(z)>\alpha_i} f_h(z) \right) \end{aligned} \tag{3.16}$$

We threat the example of the contribution of *veryLow* and *low* on the hypothesis acceptable $\mathcal{H}_A$ respectively in Figures 3.14 and 3.15:

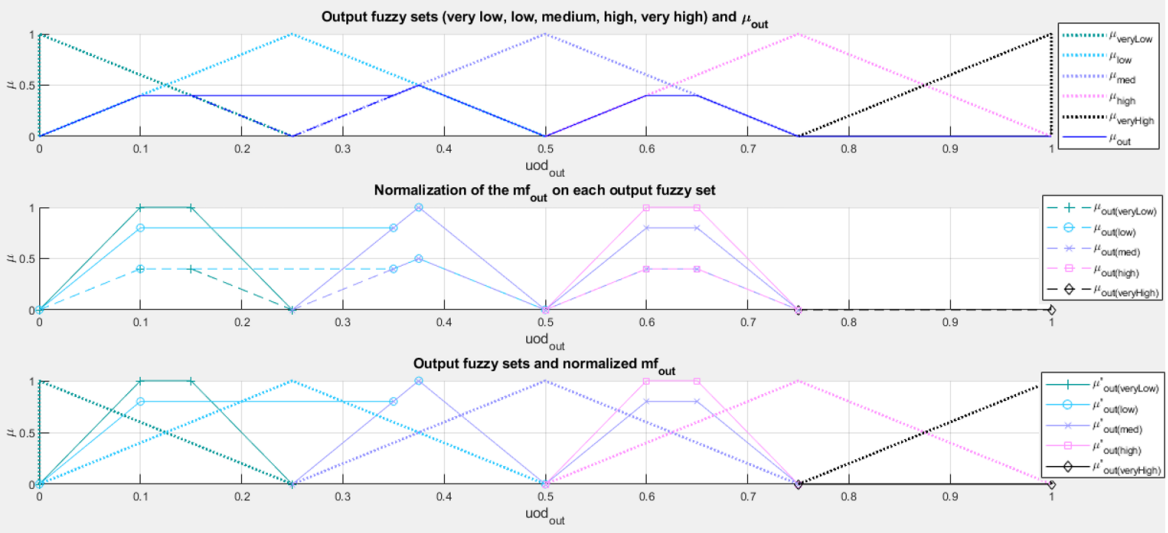


Figure 3.11: Illustration of the normalization process for the contribution of very low in dark green, low in light blue, medium in purple, high in pink and very high in black. Top: Output fuzzy sets definition in dotted lines and the resulting output μ_{out} of the combination of the C categories in solid blue line. Center: Normalization μ'_{out} in solid line of the μ_{out} in dashed lines on each output fuzzy set. Bottom: Normalized contribution of μ'_{out} in solid lines for the output fuzzy sets in dotted lines.

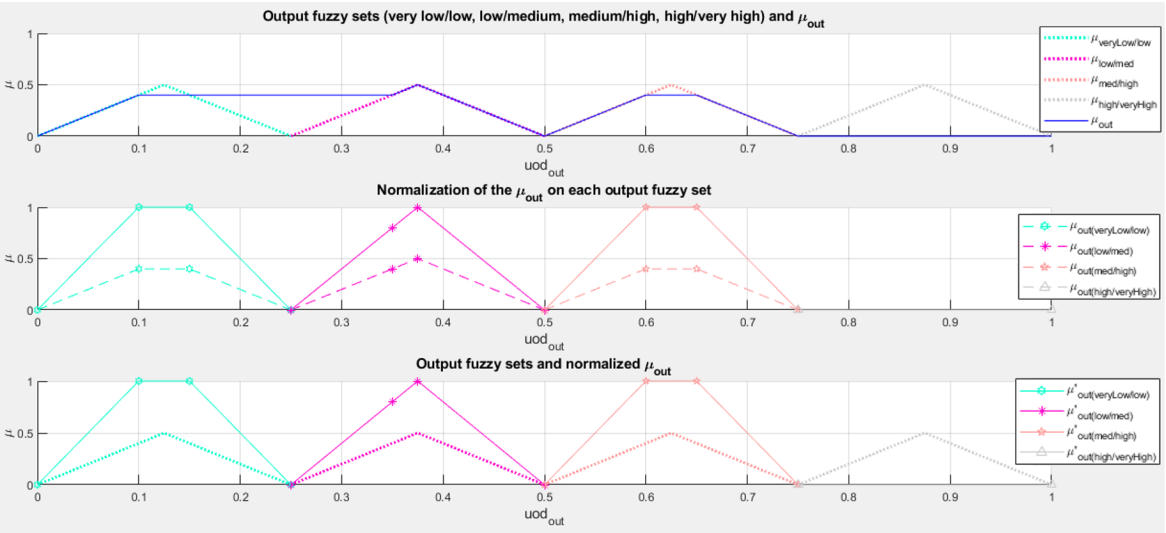


Figure 3.12: Illustration of the normalization process for the contribution of very low/low in light blue, low/medium in magenta, medium/high in light pink, high/very high in light gray. Top: Output fuzzy sets definition in dotted lines and the resulting output μ_{out} of the combination of the C categories in solid blue line. Center: Normalization μ'_{out} in solid line of the μ_{out} in dashed lines on each output fuzzy set. Bottom: Normalized contribution of μ'_{out} in solid lines for the output fuzzy sets in dotted lines.

- For the contribution of *veryLow*, in the α-set α_1, the inf value of the representation function f_A of $\mathcal{H}_A$ is equal to 1, as well as its sup value. One notices that the inf and sup values of f_A are equal to 1 for all the α-sets $\alpha_2, \alpha_3, \alpha_4, \alpha_5$. Thus, *veryLow* contributes for all the α-level subsets to the belief bel and plausibility pl of hypothesis

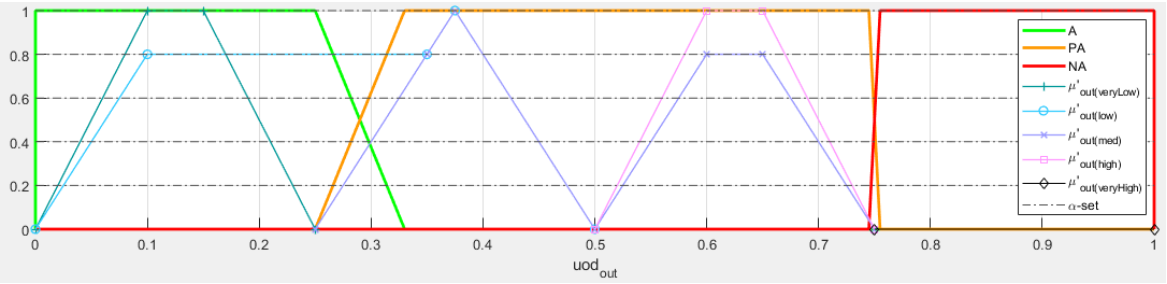


Figure 3.13: Illustration of the α-level subsets decomposition process.

$\mathcal{H}_A$ according to Equation 3.16, as summarized in Equation 3.17:

$$\begin{cases} \inf\limits_{z|\mu_{veryLow}(z)>\alpha_1} f_A(z) = 1, & \sup\limits_{z|\mu_{veryLow}(z)>\alpha_1} f_A(z) = 1, \\ \inf\limits_{z|\mu_{veryLow}(z)>\alpha_2} f_A(z) = 1, & \sup\limits_{z|\mu_{veryLow}(z)>\alpha_2} f_A(z) = 1, \\ \inf\limits_{z|\mu_{veryLow}(z)>\alpha_3} f_A(z) = 1, & \sup\limits_{z|\mu_{veryLow}(z)>\alpha_3} f_A(z) = 1, \\ \inf\limits_{z|\mu_{veryLow}(z)>\alpha_4} f_A(z) = 1, & \sup\limits_{z|\mu_{veryLow}(z)>\alpha_4} f_A(z) = 1, \\ \inf\limits_{z|\mu_{veryLow}(z)>\alpha_5} f_A(z) = 1, & \sup\limits_{z|\mu_{veryLow}(z)>\alpha_5} f_A(z) = 1. \end{cases} \tag{3.17}$$

- For the contribution of *low*, in the α-set α_1, the inf value of f_A is equal to 0, whereas its sup value is equal to 1. Consequently, *low* contributes under the α-set α_1 only to the plausibility value of hypothesis $\mathcal{H}_A$. One notices that the inf and sup values of f_A are respectively 0 and 1 for the α-sets $\alpha_2, \alpha_3, \alpha_4$. For the α_5, both the inf and sup values of f_A are equal to 0. Thus, *low* contributes for the α-level subsets $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ to the plausibility $\text{pl}(\mathcal{H}_A)$ of hypothesis $\mathcal{H}_A$, but do not contribute to its belief value according to Equation 3.16, as summarized in Equation 3.17:

$$\begin{cases} \inf\limits_{z|\mu_{low}(z)>\alpha_1} f_A(z) = 0, & \sup\limits_{z|\mu_{low}(z)>\alpha_1} f_A(z) = 1, \\ \inf\limits_{z|\mu_{low}(z)>\alpha_2} f_A(z) = 0, & \sup\limits_{z|\mu_{low}(z)>\alpha_2} f_A(z) = 1, \\ \inf\limits_{z|\mu_{low}(z)>\alpha_3} f_A(z) = 0, & \sup\limits_{z|\mu_{low}(z)>\alpha_3} f_A(z) = 1, \\ \inf\limits_{z|\mu_{low}(z)>\alpha_4} f_A(z) = 0, & \sup\limits_{z|\mu_{low}(z)>\alpha_4} f_A(z) = 1, \\ \inf\limits_{z|\mu_{low}(z)>\alpha_5} f_A(z) = 0, & \sup\limits_{z|\mu_{low}(z)>\alpha_5} f_A(z) = 0. \end{cases} \tag{3.18}$$

To conclude, an output fuzzy set will generally contribute to the bel value if its *uod* of interest (i.e. *uod* values for which $\mu \neq 0$) is included in the *uod* of interest of the representation function of the hypothesis. Its contribution to the pl value is guaranteed if its *uod* of interest intersects the *uod* of interest of the hypothesis. For example, the output fuzzy sets contributions by mf_{out} of Figure 3.9 (p. 67) to the 3 hypotheses $\mathcal{H}_{A,PA,NA}$ are depicted in Figure 3.16.

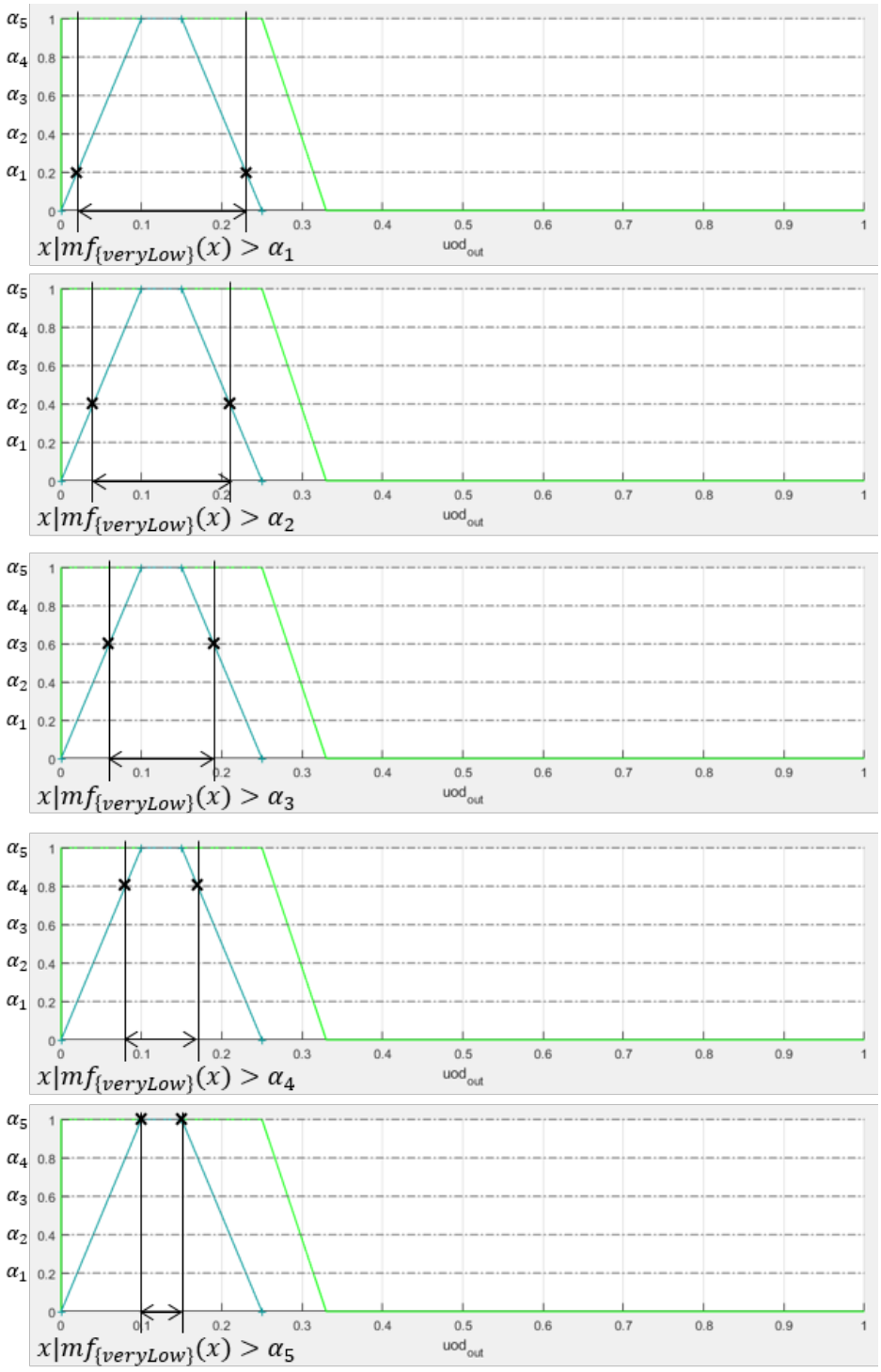


Figure 3.14: Illustration of 5 α-level subsets decomposition process to calculate the contribution of *veryLow* in dark green line to $\mathcal{H}_A$ in green line.

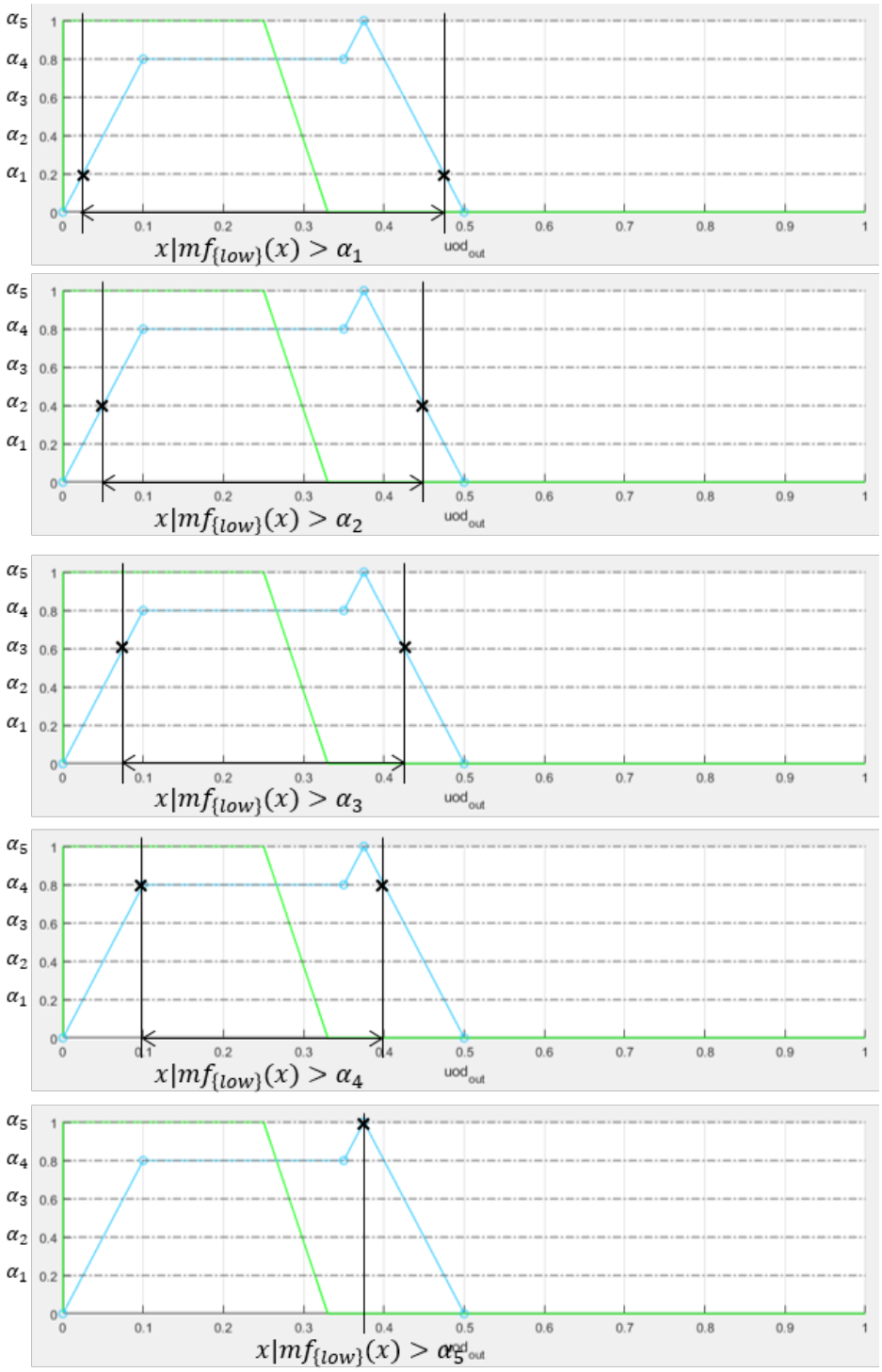
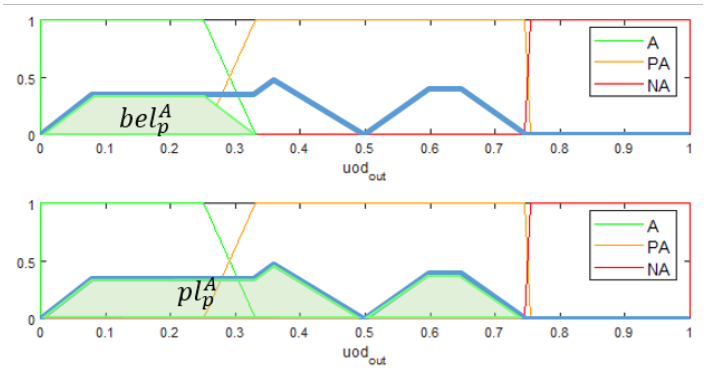
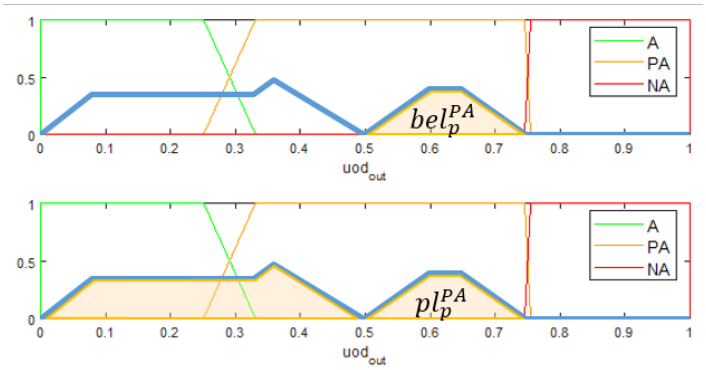


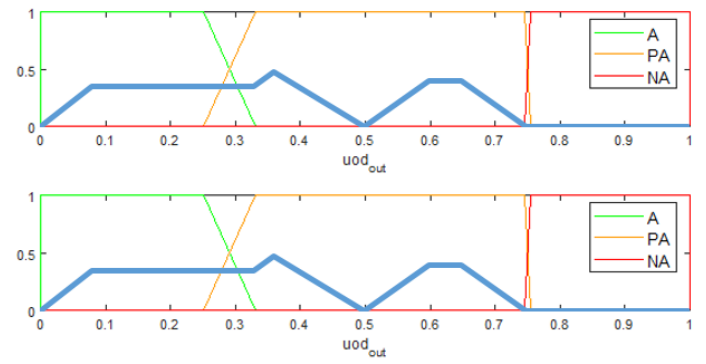
Figure 3.15: Illustration of 5 α-level subsets decomposition process to calculate the contribution of *low* in light blue line to $\mathcal{H}_A$ in green line.



(a) Contribution to hypothesis A.



(b) Contribution to hypothesis PA.



(c) Contribution to hypothesis NA.

Figure 3.16: Illustration of the belief and plausibility values for each hypothesis. (a) The small green area is the contribution to bel($\mathcal{H}_A$) and the larger green area is the contribution to pl($\mathcal{H}_A$). (b) The small orange area is the contribution to bel($\mathcal{H}_{PA}$) and the larger orange area is the contribution to pl($\mathcal{H}_{PA}$). (c) There is no contribution to bel($\mathcal{H}_{NA}$) and pl($\mathcal{H}_{NA}$) with the mf_{out} of this example.

The way to combine the length and average of the interval is part of the limits of a non-human decision process. It deals with the arbitrary choice: to favor safety or security, i.e. to assign a conservative or aggressive vehicle's behavior.

3.3.4 Level 3: Reference trajectory choice

After the implementation of level 2, we obtain for each trajectory p the risk assessment with the decision brackets $RA[\text{bel}, \text{pl}]_p^h$ as a similarity level with each hypotheses $\mathcal{H}_{A,PA,NA}$. The goal of level 3 is to return the best trajectory among a set of candidate ones $p = 1..P$, see Figure 3.3c (p. 57). As it has been highlighted in [Prokhorov 2009], it is impossible to return a precise risk quantification, but risk ranking is sufficient. The 'best' trajectory in the sense of risk assessment will then be the one with the biggest/smallest value of a risk indicator.

As previously stated, two pieces of information must be interpreted in the decision brackets $RA[\text{bel}, \text{pl}]_p^h$: (i) the smaller the interval is, the stronger the knowledge on the probability of being in this state is, (ii) the closer the interval is to 1, the greater the probability of being in the state is. Thereby, we propose a risk indicator $\mathcal{I}_p$ for each trajectory p considering both the mean value $\bar{\mathcal{H}}_h$ and the length $l(\mathcal{H}_h)$ of the associated belief (bel)-plausibility (pl) decision brackets $RA[\text{bel}, \text{pl}]_h^p$. $\bar{\mathcal{H}}_h$ and $l(\mathcal{H}_h)$ are defined in Equation 3.16 with bel $\leq$ pl:

$$\begin{aligned} \bar{\mathcal{H}}_j &= \frac{(\text{pl}(\mathcal{H}_j) + \text{bel}(\mathcal{H}_j))}{2}, \\ l(\mathcal{H}_j) &= \text{pl}(\mathcal{H}_j) - \text{bel}(\mathcal{H}_j). \end{aligned} \tag{3.19}$$

In the work proposed by [Furda & Vlacic 2010], the authors calculate the resulting value to define the best driving maneuver with the Simple Additive Weighting Method. They previously define a utility degree and importance weight to each maneuver. By analogy, we consider the utility degree as the mean value of the decision brackets and the weight of that decision as the inverse of the interval length. That is to say, the higher the mean value is within the smaller brackets, the better and more accurate the risk assessment is. Moreover, we affect $\mathcal{H}_h$ a weight Ω_h so that the acceptability and non-acceptability values stand out in the decision process. The risk indicator for the p^{th} trajectory is expressed in Equation 3.20:

$$\mathcal{I}_p = \sum_h \Omega_h \frac{\bar{\mathcal{H}}_h}{\max(l(\mathcal{H}_h), \varepsilon)} \tag{3.20}$$

To avoid zero-division and to keep consistency on the indicator, we take the maximum between the interval length and an ε value. The choice of Ω_h must respect the intuitive reasoning saying that (i) the NA value is discriminating, (ii) the A value is more significant than the PA value, and (iii) if all the hypotheses are the same, the risk indicator value is ordinary, i.e. close to 0.

3.4 Simulation results

In order to validate our decision-maker, we propose a simple case study in simulation. More complex highway use cases are presented in experimentation in Chapter 5. We use sigmoid candidate trajectories as inputs in this simulation. The case study is described in subsection 3.4.1, the parameters for FDST are set in subsection 3.4.2, and the simulation results are depicted in subsection 3.4.3.

3.4.1 Case study

In this application, we consider a 2-lane highway, with one obstacle going straight on the right lane at speed 80 km/h and 2 s ahead of the ego vehicle at speed 100 km/h. We choose 7 acceleration profiles based on no acceleration, 3 acceleration levels, and 3 deceleration levels to create tentacle trajectories in the form of straight line for lane following and sigmoid for lane change. The equation of the sigmoid is expressed in Equation 3.21, with y the lateral coordinate and x the longitudinal one:

$$y(t) = y_0 \pm \frac{b}{1 + e^{-\lambda(x(t)-c)}}, \tag{3.21}$$

with b the shift between y_0 and the asymptotic end value, which corresponds in our problem to the target centerline. The sigmoid parameter λ and the distance delay c are tuning parameters. In our problem, they are used for driver safety and comfort, which depend on the lateral and longitudinal ego vehicle's velocities (see subsection 4.3.1.3, p. 90 for more details).

We obtain 14 candidate trajectories: 7 for lane changing *Left* and 7 for car following maneuvers *Straight*, see Figure 3.17.

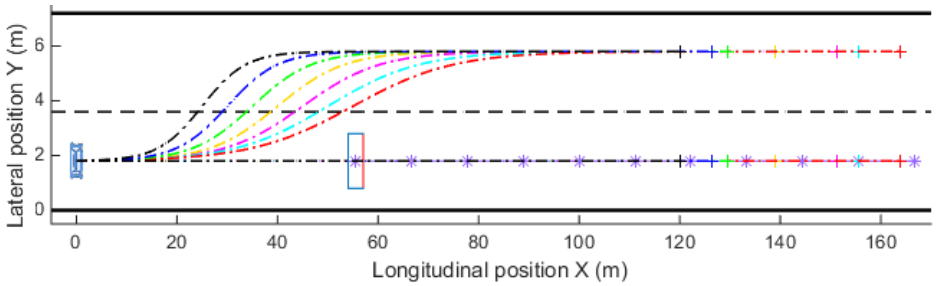


Figure 3.17: Case study description. The ego vehicle is represented in blue. The obstacle is the blue rectangular with the red front line. The 7 acceleration profiles give 14 dotted-dashed trajectories for left and straight directions: no acceleration $p1$ in yellow, $p2 : \frac{a_{x,max}}{4}$ in magenta, $p3 : \frac{a_{x,max}}{3}$ in cyan, $p4 : \frac{a_{x,max}}{2}$ in red, $p5 : \frac{d_{x,max}}{4}$ in green, $p6 : \frac{d_{x,max}}{3}$ in blue, $p7 : \frac{d_{x,max}}{2}$ in black, with respectively $a_{x,max} = 2$ m/s^2 and $d_{x,max} = -1.5$ m/s^2 the maximum acceleration and deceleration of the ego vehicle. The ends of trajectories are marked with a plus sign. The obstacle trajectory is the dotted-asterisk purple line.

Table 3.3 gives the parameters of the left trajectories $p = 1..7$.

Table 3.3: Trajectory parameters for the 7 left lane changes.

Trajectory	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
Acceleration (m/s^2)	0	0.5	0.66	1	-0.375	-0.5	-0.75
λ (m^{-1})	0.15	0.13	0.11	0.10	0.17	0.20	0.23
c (m)	40	45	50	55	35	30	25

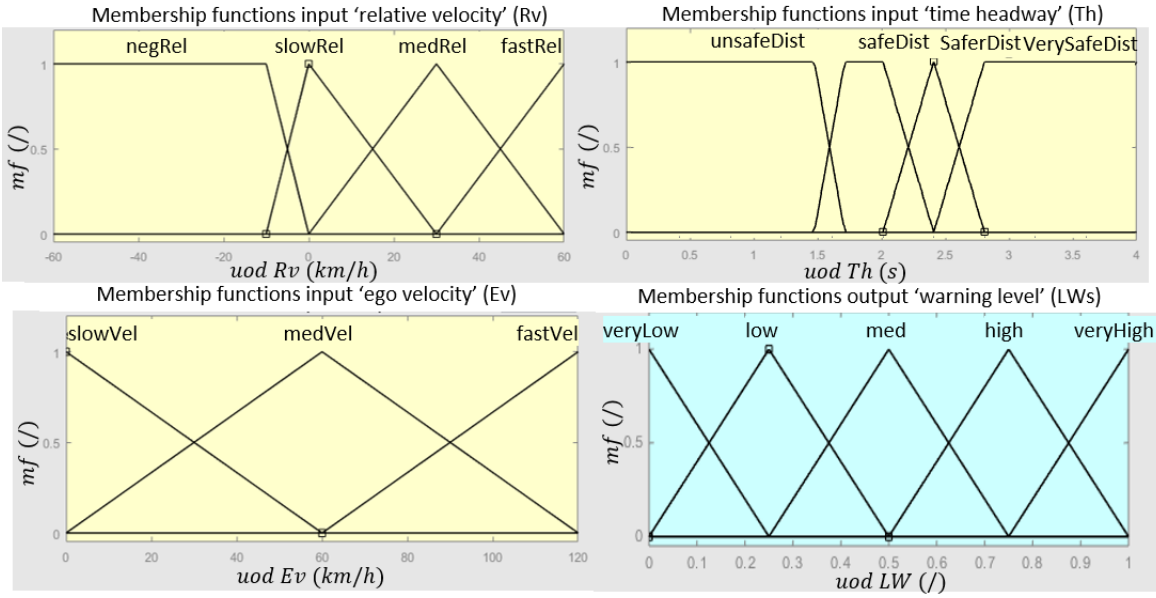
3.4.2 Parameters

We detail in this part the 2 categories used for this use case: obstacle safety ($\kappa = 3$) and driving rules ($\kappa = 4$), see Table 3.1. The membership functions and rule-base of category 3 – Obstacle safety are adapted from [O' Brien *et al.* 2017]. The authors used the dataset NGSIM and the Insurance Corporation of British Columbia's (ICBC) driving rules [ICB 2015] to create the input fuzzy sets for the criteria 'Relative velocity' (Rv in km/h), 'Time headway' (Th in s) and 'Ego velocity' (Ev in km/h), and the 'warning level' output fuzzy sets $LW_3^{\iota \in \{veryLow, low, med, high, veryHigh\}}$. The μ are plotted in Figure 3.18a and the rule-base is written in Table 3.4. As we consider in this paper an autonomous vehicle, we shift 1 s less the membership functions of the 'Time headway' criteria, which was initially designed in [O' Brien *et al.* 2017] for the analysis of human drivers. This is justified by the 1 s assessed to the driver reaction time.

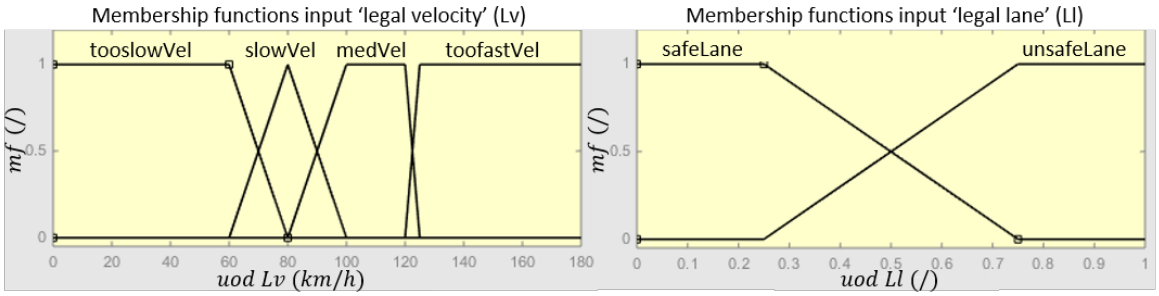
Figure 3.18b shows the FIS for category 4, based on [ICB 2015]. The input fuzzy sets are 'Legal velocity' (Lv in km/h) and 'Legal lane' (Ll no unit), and the output fuzzy sets are the same 'warning level' as for category 3. We use a *Mamdani* FIS with a minimum AndMethod to the set of rules, and respectively the minimum, maximum, for implication ORR, aggregation ARR, methods.

In order to show the influence of the trust and weight consideration, we present 3 scenarios. Scenario 1 is a nominal situation with sensors' trust values set to 100%. Scenario 2 presents a nominal situation with a 30% trust value for 'Time headway' and 100% for the two other criteria of category 3. Lastly, scenario 3 keeps the trust values of scenario 1 in critical situation, where category 3 is hard and category 4 middle. In nominal situation, we consider the hard, middle, and soft weights as equivalent, i.e. $w_{hard,middle} = 1$ for our 2 categories 3 and 4. In critical situation, we consider hard weight ten times bigger than middle weight. For conjunctive combination, we use the inverse of the weight value in the min operator, whereas for disjunctive combination, we use the weight value in the max operator. As we apply the conjunctive Yager's rule, we get $w_{hard} = 1/10$ and $w_{middle} = 1$.

The 3 hypotheses for risk assessment with respectively Acceptable hypothesis (A), Partially Acceptable (PA), and Non-Acceptable (NA) are plotted respectively in green, orange, and red in Figure 3.8 (p. 65). We choose the hypotheses weights $\Omega_A = 2$, $\Omega_{PA} = 1$ and $\Omega_{NA} = -3$ for the risk indicator calculation, and $\varepsilon = 0.1$ for consistency, as belief and plausibility values expressed on $[0, 1]$. The highest $\mathcal{I}_p$ leads to the best trajectory p.

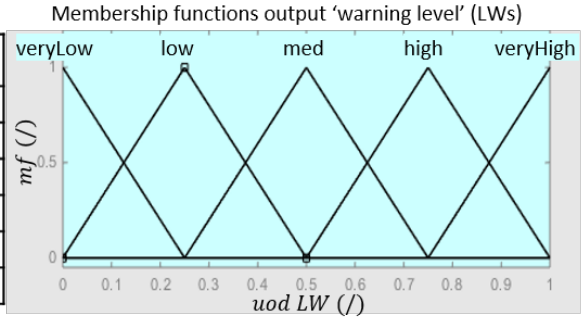


(a) Fuzzy Inference System for category 3 – Obstacle safety.



Fuzzy Logic rule base

N°	Legal Velocity	Legal Lane	Warning
1	tooslowVel	/	veryHigh
2	toofastVel	/	veryHigh
3	slowVel	safeLane	med
4	slowVel	unsafeLane	high
5	medVel	safeLane	low
6	medVel	unsafeLane	veryLow



(b) Fuzzy Inference System for category 4 – Driving rules.

Figure 3.18: FIS for categories 3 and 4.

Table 3.4: Rule-base for category 3 – Obstacle safety.

Rule Number	Relative Velocity	Time Headway	Velocity	Warning
1	slowRel	unsafeDist	slowVel	high
2	slowRel	unsafeDist	medVel	high
3	slowRel	unsafeDist	fastVel	veryHigh
4	slowRel	safeDist	slowVel	veryLow
5	slowRel	safeDist	medVel	low
6	slowRel	safeDist	fastVel	med
7	slowRel	SaferDist	slowVel	veryLow
8	slowRel	SaferDist	medVel	veryLow
9	slowRel	SaferDist	fastVel	low
10	slowRel	VerySafeDist	slowVel	veryLow
11	slowRel	VerySafeDist	medVel	veryLow
12	slowRel	VerySafeDist	fastVel	veryLow
13	medRel	unsafeDist	slowVel	high
14	medRel	unsafeDist	medVel	veryHigh
15	medRel	unsafeDist	fastVel	veryHigh
16	medRel	safeDist	slowVel	med
17	medRel	safeDist	medVel	med
18	medRel	safeDist	fastVel	med
19	medRel	SaferDist	slowVel	low
20	medRel	SaferDist	medVel	low
21	medRel	SaferDist	fastVel	low
22	medRel	VerySafeDist	slowVel	veryLow
23	medRel	VerySafeDist	medVel	veryLow
24	medRel	VerySafeDist	fastVel	veryLow
25	fastRel	unsafeDist	slowVel	veryHigh
26	fastRel	unsafeDist	medVel	veryHigh
27	fastRel	unsafeDist	fastVel	veryHigh
28	fastRel	safeDist	slowVel	high
29	fastRel	safeDist	medVel	high
30	fastRel	safeDist	fastVel	veryHigh
31	fastRel	SaferDist	slowVel	med
32	fastRel	SaferDist	medVel	med
33	fastRel	SaferDist	fastVel	high
34	fastRel	VerySafeDist	slowVel	low
35	fastRel	VerySafeDist	medVel	med
36	fastRel	VerySafeDist	fastVel	high
37	negRel	unsafeDist	slowVel	med
38	negRel	unsafeDist	medVel	high
39	negRel	unsafeDist	fastVel	veryHigh
40	negRel	safeDist	slowVel	veryLow
41	negRel	safeDist	medVel	low
42	negRel	safeDist	fastVel	low
43	negRel	SaferDist	slowVel	veryLow
44	negRel	SaferDist	medVel	veryLow
45	negRel	SaferDist	fastVel	veryLow
46	negRel	VerySafeDist	slowVel	veryLow
47	negRel	VerySafeDist	medVel	veryLow
48	negRel	VerySafeDist	fastVel	veryLow

3.4.3 Results

The criteria values obtained with the corresponding predictive operator (Table 3.1) for our candidate trajectories are collected in Table 3.5. Considering the input membership functions of categories 3 and 4, it seems that for category 3, the 'Time headway (Th)' criterion would be the most differentiated one, whereas for category 4, 'Legal velocity (Lv)' and 'Legal lane (Ll)' would both influence the degree of membership on the output warning risk levels LW.

Table 3.5: Criteria values for categories 3 and 4.

	Left trajectories							**Straight trajectories**						
	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
Rv	20	29	32	38	20	20	20	20	29	32	38	20	20	20
Th	1.73	1.62	1.54	1.45	1.81	1.84	1.89	1.01	0.52	0.38	0.11	1.44	1.59	1.79
Ev	100	104.5	106	109	96.625	95.5	93.25	100	104.5	106	109	96.625	95.5	93.25
Lv	100	109	112	118	93.25	91	86.5	100	109	112	118	93.25	91	86.5
Ll	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0

Table 3.6 shows the risk indicator $\mathcal{I}_p$ of each straight and left trajectories $p = 1..7$ for the 3 scenarios.

Table 3.6: Risk indicator $\mathcal{I}_p$.

	Left trajectories							**Straight trajectories**						
	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
Scenario 1	3.86	3.90	3.86	3.77	3.37	3.38	3.44	3.37	0.52	0.28	0.28	3.57	3.76	3.85
Scenario 2	3.97	3.71	3.61	3.61	3.64	3.73	3.87	3.69	0.71	0.37	0.37	3.42	3.58	3.99
Scenario 3	3.89	4.09	4.19	4.21	3.12	3.05	2.97	3.43	0.52	0.28	0.28	3.49	3.58	3.33

For **scenario 1**, the left candidate trajectories scores are close and positive, as their criteria values through the rule-base mostly lead to low, medium and high risk levels. For the straight trajectories, collision with the front obstacle is very close for the acceleration profiles as shown in Figure 3.17, where the obstacle trajectory's (purple dotted-asterisk line) distances with $p2$, $p3$, and $p4$ trajectories (resp. magenta, cyan and red plus-sign lines) are less than 15 m. This potential collision is expressed with Th criterion smaller than 1 s. On the other side, straight trajectory $p1$ has a lower score than left trajectory $p1$ because of a smaller Th and category 4 yielding low risk level for straight lane and veryLow for left one. On the contrary, straight $p5$, $p6$, and $p7$ have a better score than left trajectories as they remain on the right lane which is safer than changing lane at low velocities. The decision brackets are depicted in Figure 3.20a. For each hypothesis, the $[\text{bel}, \text{pl}]$ interval for the 7 left trajectories and 7 straight trajectories are detailed. The analysis of the beliefs intervals for all the trajectories confirms that the most pronounced risk assessment is for acceptable (A) and partially acceptable (PA) hypotheses. If we look at the rule-base, category 3 induces mostly medium, high and veryHigh warning risk levels and category 4 veryLow and low levels. Thus the similarity with PA hypothesis is higher than the one with A or NA hypotheses. The decision brackets for straight trajectories $p2$, $p3$ and $p4$ are very close or set to 0, as previously explained. Only the deceleration trajectories cover the non-acceptable hypothesis (NA), as category 4 yields med and high risk levels, i.e high and veryHigh LW. Considering criteria values of category 4, the best trajectory is among the left medium velocities. The values

for category 3 are less discriminative. The best trajectory for scenario 1 in the sense of the previously defined risk indicator is then Left $p2$.

During **scenario 2 - lower trust**, the lower trust value influences the aggregate output membership function, as shown in Figure 3.19 for the example of the candidate trajectory left with no acceleration $p1$. We notice that influences of trust in 'Time headway' for category 3 is important for all the warning levels. The corresponding belief intervals are displayed in Figure 3.20b. We notice that the length of the intervals remains almost the same for all the candidate trajectories under all hypotheses, but the mean values are smaller for A and PA hypotheses. Indeed, the Th criterion mostly leads category 3 to veryLow, low and medium risk levels in this case study. If the criterion is being less considered as its trust is lower, the *bba* for those values decrease. Besides, the mean values of A and PA hypotheses are closer than in scenario 1. The best trajectory for scenario 2 is to go Straight $p7$.

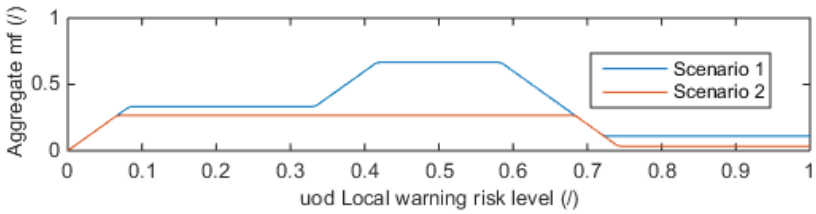
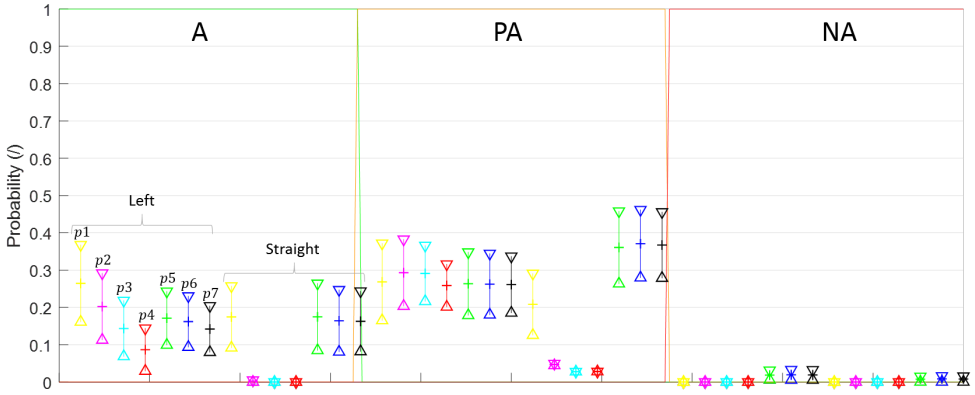


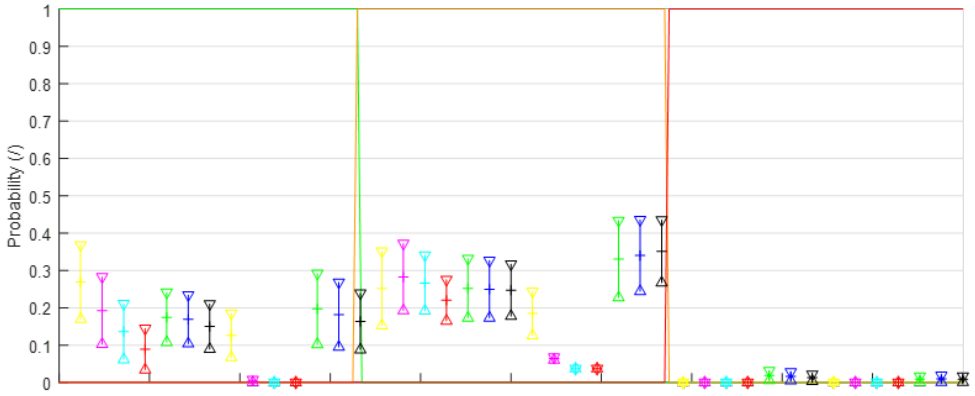
Figure 3.19: Influence of trust values on the aggregate function of category 3 for the left trajectory $p1$.

In **scenario 3 - critical situation**, the corresponding belief intervals are displayed in Figure 3.20c. We first observe that there is no change on NA hypothesis, which proves that category 3 has more critical weight on the safety of the ego-vehicle. If we analyze the rule-base of category 3 in details, we notice that the faster the ego velocity is, the riskier the situation is. In the same way, the bigger the relative velocity is, the higher the risk is. On the other side, the bigger the time headway is, the safer the situation is. Thus we notice that the intervals for the acceleration profiles $p2, p3, p4$ have smaller length with the same mean value for A and higher mean value for PA. The opposite situation occurs for the deceleration profiles $p5, p6, p7$. At the end, the best trajectory for scenario 3 is to go Left with high acceleration $p4$.

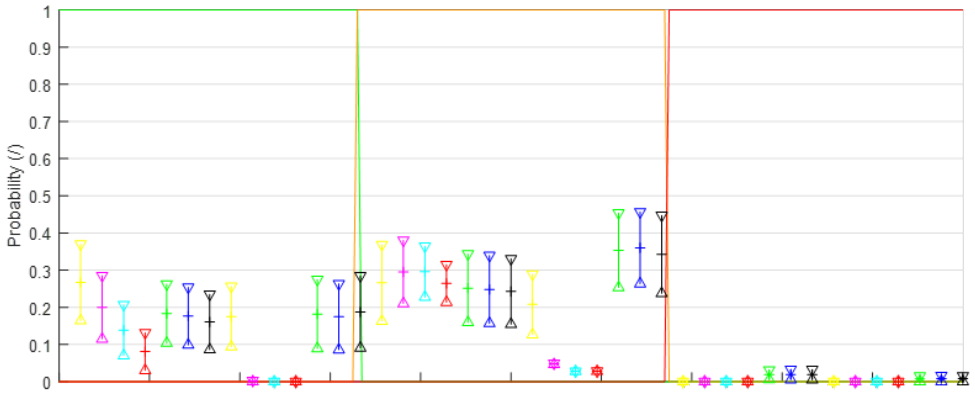
Finally, we notice that (i) for most cases, the decision brackets are large, which means that the fuzzyfication is important and the hypotheses may thus be too large, (ii) with close candidate trajectories, the tested categories return similar results for the 3 scenarios, (iii) the results are sensitive to rules-base consistency, (iv) the values for the risk indicator are relevant in the 3 scenarios, and (v) considering criteria trust values and weighted categories shows sensible changes on the risk assessment. In fact, to observe the respective influence of trust value and weight category, the untrusted criteria must influence the LW_c^i assignment, and the overlap of the LW_c^i between two categories must coincide with the hypotheses representation functions transition.



(a) Decision brackets for scenario 1.



(b) Decision brackets for scenario 2.



(c) Decision brackets for scenario 3.

Figure 3.20: For each of the hypotheses (A) in green, (PA) in orange and (NA) in red, the decision brackets [bel, pl] of each candidate trajectory are plotted.

3.5 Conclusion and future work

We presented a new framework for predictive multi-criteria decision making for autonomous vehicles, which simultaneously considers several factors of risk, their uncertainties, and their relative importance for human safety. We employed the belief theory for risk assessment and fuzzy logic for uncertain data with a trustworthiness value. As the fuzzy rules are based on both traffic rules and real data learning, the decision relies on a conservative but active behavior of the ego vehicle. Moreover, the Fuzzy Dempster-Shafer reasoning avoids both the defuzzification process for fuzzy logic by using belief theory and the problem of fallacy of the excluded middle for belief theory by using fuzzy set and appropriate rule-base. We also introduced the influence of sensors trust values and relative importance on criteria and categories to relax if safer or to consider driver preferences to get a flexible decision. Another advantage is the possibility to evaluate any description of motion, i.e. paths, trajectories, or symbolic maneuvers, as long as they can be characterized through a set of criteria. Lastly, the proposed framework is scalable and ables to evaluate arbitrary functions, although it requires fine tuning for fuzzy inference systems and best trajectory definition to respect the problem's specification. However, the combination of fuzzy logic and evidential reasoning still depends on a mathematically tuning formalism for the fuzzy membership functions, fuzzy rules and the hypothesis definition. It also suffers from an explicit mathematical function to simply expressed the decision rules.

Future work will consist in deploying this approach to other use cases with more risk factors based on more criteria and/or categories. In order to adapt the behavior of the fuzzy inference systems to a human-in-the-loop behavior, learning techniques can also be considered for different driver profile datasets to form the fuzzy sets. Lastly, if the criterion leading to an unsafe decision is identified, transposing it into a safe universe will allow to apply preventive decisions.

After defining the decision making function, one needs to provide the candidate motions to evaluate as the best one for the reference motion sent to the controller. This is the research topic of Chapter 4.

Chapter 4

Optimized Trajectory Planning within Non-Collision Nominal Intervals

This chapter considers the problem of near-optimal trajectory generation for autonomous vehicles on highways. The goal is to select a predictive reference trajectory in the free evolution space, while avoiding both generating a pre-calculated set of candidate trajectories and decoupling path and velocity optimizations. Moreover, this trajectory aims at optimizing a decision process based on multi-criteria functions, which are not straightforward to design and can have a black-box formulation. The main idea is to use the decision evaluation function in the trajectory generator with a Simulated Annealing approach. The parameters of a sigmoid trajectory are optimized within Non-Collision Nominal Intervals, which are defined as collision-free intervals under nominal conditions using a velocity-space representation.

Contents

The introduction of autonomous vehicles into a human environment involves uncertainties and complex behaviors. In this respect, motion planning algorithms are given a greater role in the autonomous scheme, stated as an intelligent linker between sensors and actuators. As it has been detailed in subsection 2.1.2 (p. 15), the motion planning block is commonly decomposed into the five following functions: route planning, obstacles prediction, motion generation, decision making, and motion deformation. Our work focuses on the functions *motion generation* and *decision making* (previously exposed in Chapter 3). In the literature, authors usually treat these two subparts separately and sequentially, either by first generating candidate motions and then evaluating the most appropriate one according to the objective function of a decision algorithm, or by first making the decision on the most appropriate maneuver and then generating a motion to fit properly. Our contribution addresses both the *decision making* –as an evaluation function– and the *motion generation* –under the form of trajectory generation– in a combined manner, in order to ensure the consistency between the choice of a maneuver and the calculation of the trajectory.

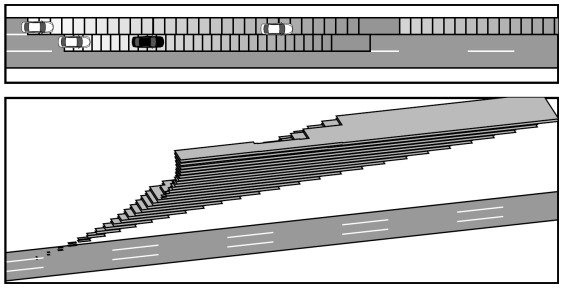
This chapter is organized as follows: Previous work and motivation are presented in section 4.1. The new architecture is presented in section 4.2. Section 4.3 describes the reachable space-time with the problem description and the NCNI interpretation. The proposed algorithm is detailed in section 4.4. The SA-optimization method is explained in section 4.5. Lastly, a numerical example is discussed in section 4.6, and section 4.7 presents the conclusion and future work.

4.1 Literature review and motivation

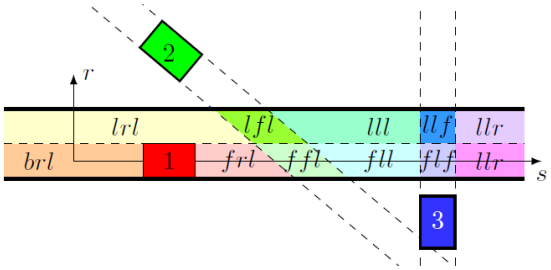
Two approaches are distinguished for trajectory generation in autonomous vehicles: *discretization* and *decoupling*. The first one generates a set of candidate trajectories called tentacles, introduced in [Von Hundelshausen *et al.* 2008], as further described in subsection 2.2.2.4 (p. 31). The generation is made a priori, based on predefined geometric curves, such as line and circle [Cherubini *et al.* 2012], clothoid [Chebly *et al.* 2017] or sigmoid [Cesari *et al.* 2017]. The completion of each candidate trajectory is then tested and scored against the objective function of the decision subpart in order to define the most appropriate one. The second approach is based on static and dynamic decoupling. A spatial path solution is first defined on a static decomposition of the space, for example fitting particular points with polynomials [Xu *et al.* 2012] or Bézier curves [Lattarulo *et al.* 2017]. This representation uses the topography of the road, such as lane marking, road shape, map information, and static obstacle avoidance. The path is then adjusted to the dynamic obstacles by choosing a speed/acceleration profile [Liu *et al.* 2017b] to respect the speed limits and avoid dynamic obstacles. Conversely, defining first the optimal velocity profile dynamically and then the static path is also possible [Wang *et al.* 2018b].

From these two representations emerge two main issues. Indeed, the *discretization* of the candidate solutions requires either testing a large number of candidate profiles to approach the optimal one, or to make a choice a priori on the characteristics of the solution. This problem is studied with the continuous optimization strategies mentioned in subsection 2.2.2.5 (p. 33), such as gradient descent [Xu *et al.* 2012], Linear Programming (LP) [Plessen *et al.* 2017], Mixed-Integer Programming (MIP) [Wang *et al.* 2018b], or Model Predictive Control (MPC) [Cesari *et al.* 2017]. The drawback of such strategies is their need for an explicit objective function. Similarly, the *decoupling* approach does not allow exploring all motions over the space-time prediction horizon, as it optimizes each dimension separately. This has previously been addressed through the Velocity Obstacles (VO) representation [Fiorini & Shiller 1998], the Inevitable Collision States (ICS) [Fraichard & Asama 2004], reachable sets [Söntges & Althoff 2018] (see Figure 4.1a), or the graph interval formulation [Altché & De La Fortelle 2017] (see Figure 4.1b). By combining these ideas, we propose the Non-Collision Nominal Intervals (NCNI), which return, based on a velocity representation, the predictive collision-free intervals for nominal maneuvers. NCNI are inspired from the complementary space of ICS, and are characterized as spatiotemporal intervals in order to define the reachable space-time for navigation and the corresponding graph of maneuvers.

The last motivation which drives our research is the optimization of the trajectory, evaluated with the previously developed decision-maker in Chapter 3. The main issue is the optimization of the objective function that uses the Fuzzy Dempster-Shafer Theory (FDST). However, this function can not be expressed explicitly as a simple mathematical function. Indeed, we consider our decision-maker as a black-box function. Moreover, we want to treat a continuous solution space in order to tackle the two main issues from trajectory generation mentioned in the previous paragraphs. Thus, there exists a huge number of potential solutions. Black-box objective functions with a huge number of potential solutions and no a priori knowledge are usually optimized using metaheuristics [Gendreau & Potvin 2010]. The advantages of metaheuristics approaches are that they do not require a formalized problem, and do not suffer from combinatorial explosion. Moreover, the heuristic property tackles the



(a) Reachable sets representation from [Söntges & Althoff 2018]. Up: Constant velocity prediction of the three surrounding obstacles. Down: The grey area corresponds to the free-space where the ego vehicle can navigate.



(b) Partitioning space from [Altché & De La Fortelle 2017]. The colored space indicates the relative position of the ego vehicle respectively to obstacles 1, 2 and 3 (b=behind, f=front, l=left, r=right).

Figure 4.1: Illustration of the reachable sets and graph interval with a partitioning space.

local minima issue. Nevertheless, the main drawbacks of these methods are the need for a fine problem-adaptive tuning, their non-deterministic resolution, and a trade-off between optimality, completeness and execution time. In order to choose the most adapted algorithm among the large variety of metaheuristics approaches, we list our problem properties, based on the decision-maker developed in Chapter 3, and the motivations for the trajectory planner:

- our objective function is a black-box function: the algorithms based on an explicit resolution of the objective function such as the gradient approaches are dismissed;
- the parameters to optimize are non-independent: the population-based algorithms, e.g. Genetic algorithm or Particle Swarm Optimization (PSO), are not exploited;
- there is no a priori on the function, which can have local minima: a solution to escape local minima is required (even if a local minimum might be accepted);
- the algorithm is required to run in real-time: the solutions with memory, e.g. Tabu search, are dismissed.

As reviewed in [Wolpert & Macready 1997], the Simulated Annealing (SA) algorithm tackles these properties issues, while guaranteeing theoretical convergence.

The strategy that we propose is the combination of NCNI and SA algorithm to address the discretization and decoupling issues. This approach results in a new architecture for trajectory generation and decision making, detailed in the next section.

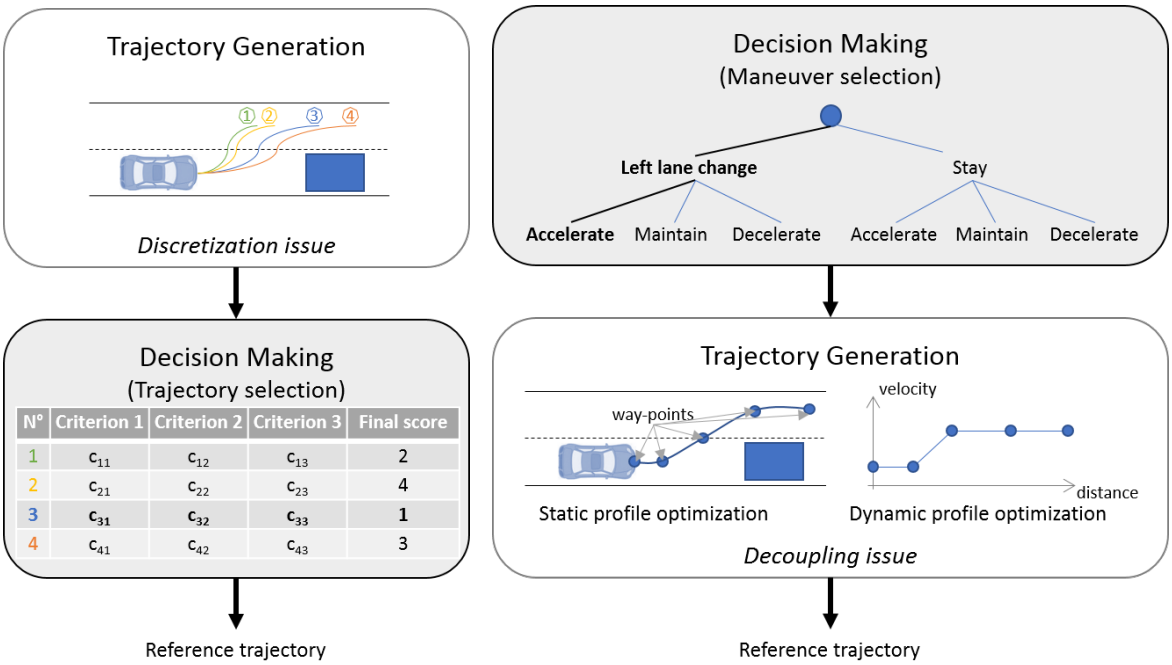
4.2 All-in-one motion planning architecture

As presented in the previous section, two strategies occur for combining motion generation and decision making, which are the two functions in the scope of our work.

In the first strategy represented in Figure 4.2a, the trajectory (or more generally motion) generation is first addressed, and then the decision making. To do so, the evolution space

is usually discretized in order to propose predefined trajectories. Thus, the decision making function acts as a trajectory selector among the previously predefined trajectories to return the one which has the best score following some criteria. For example in Figure 4.2a, four predefined trajectories (considering both position and velocity) are drawn. The decision-maker is based on three criteria to select the final choice. As trajectory 3 has the best score, it becomes the reference trajectory to follow by the controller.

In the second strategy represented in Figure 4.2b, the decision making is first applied, and then the trajectory generation. The decision-maker considers a maneuver selection with an orientation and a velocity profile. In the case of highway driving, the maneuver can be reduced to stay on its lane or a left/right lane change, and the velocity profile to maintain speed, accelerate or brake. A trajectory is then generated in the evolution space, which corresponds to the selected maneuver. To do so, the trajectory is usually decoupled in a static profile (position) and a dynamic one (velocity). This two-dimension optimization problem is solved by optimizing first the static profile and then the dynamic one, or vice-versa. For example in Figure 4.2b, the decision-maker selects a left lane change with acceleration. The five corresponding way-points are generated to define the maneuver in static, and a velocity profile with acceleration is optimized for the dynamic part of the maneuver. Both static and dynamic parts form the reference trajectory.



(a) Architecture with discretization issue.

(b) Architecture with decoupling issue.

Figure 4.2: Illustration of the two classic architectures for motion generation and decision making.

In this work, we introduce an all-in-one architecture, in which the motion generation is jointly optimized with a pre-existing objective function $\mathcal{J}_{obj}$ from the decision-maker, see Figure 4.3. First, Non-Collision Nominal Intervals (NCNI) are calculated from the interpretation of the environment data (road, obstacles, ego vehicle). They are defined as a continuous

reachable space-time, taking into account the kinematic constraints of the ego vehicle and the collision-free states (position and velocity) for nominal highway driving conditions. Then, maneuvers are considered if there exists a common space-time between two NCNI. For each maneuver, a parameterized trajectory is generated within the NCNI. This trajectory is evaluated according to the decision-maker previously detailed in Chapter 3. The optimization process is applied to the parameters of the trajectory to find the near-optimal trajectory for each maneuver. Lastly, each near-optimal trajectory of each maneuver are compared and the one with the best score according to the objective function $\mathcal{J}_{obj}$ of the decision-maker is returned as the reference trajectory sent to the control block.

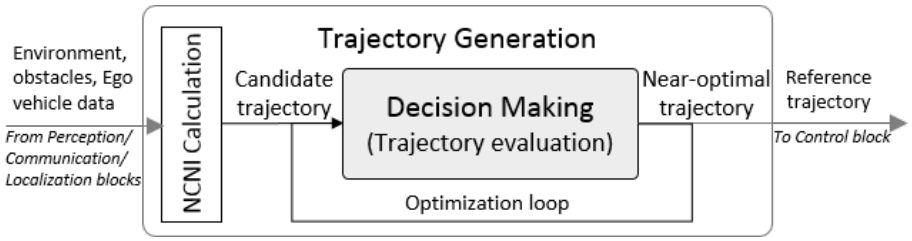


Figure 4.3: All-in-one motion planning architecture.

4.3 The reachable space-time as NCNI

The reachable space-time is defined as the evolution space fulfilling the following conditions: (i) it must be consistent with the physical limitations of the vehicle, (ii) it is represented with position and velocity information, (iii) it is predicted along a space and time horizon, and (iv) it is collision-free over the prediction horizon or, if a collision has been predicted along the prediction horizon, it can be avoided through a nominal maneuver, e.g. braking or a lane changing. The environment is described in subsection 4.3.1 and the NCNI in 4.3.2.

4.3.1 Environment description

The previous definition of the reachable space-time requires first a description of the evolution space, which mainly corresponds to the road model in the case of the highway environment (subsection 4.3.1.1). Then, the specification of the vehicle states (subsection 4.3.1.2) is needed to define the trajectory model (subsection 4.3.1.3). Finally, in order to navigate in a collision-free space, a prediction of the obstacles' behaviors is necessary (subsection 4.3.1.4). This section details all the models used in this work.

4.3.1.1 Road model

The reachable part of the road is delimited by the traffic lanes and the legal speed limits (v_{min}, v_{max}). In order to limit the calculation time, only the ego lane and the adjacent left and right lanes are modeled for the future motion. Indeed, we consider, under nominal behaviors, only one lane change at a time due to the limitation of the perception area. However, there is no theoretical limitation for modeling more lanes in the algorithm process. This limitation

of the first adjacent lanes only values for the lane change behavior modeling, more lanes can be considered to locate the ego vehicle on the road for route planning.

In our work, we use a road-aligned coordinate frame under the approximation of straight lines for highways (the radius of curvature is higher than 1000 m [Reif & Dietsche 2010]). Nevertheless, different methods exist to transform the road coordinate from a global frame to a road-aligned coordinate frame, for example authors in [Altché & De La Fortelle 2017] apply a local Frenet frame, and authors in [Hudecek & Eckstein 2014] first decompose the road into the three segments straights, curves and arcs, and then transform the segment-aligned world coordinates into segment-aligned road coordinates.

4.3.1.2 Vehicle model

The use of a simple model simplifies the consideration of the constraints with a parameterized curve and allows low time calculation. However, it might result in inconsistent reference trajectories with the real vehicle dynamics. The use of more complex models provide more realistic vehicle dynamics. In the scope of motion planning, authors in [Polack *et al.* 2017] give a condition on the lateral acceleration to validate the use of the kinematic bicycle model for planning purposes. As the control part is out of the scope of this thesis work, we use the simplest particle kinematics to model the ego vehicle dynamics. We note $\chi = (\ddot{x}, \dot{x}, x, \ddot{y}, \dot{y}, y, \alpha)$ the state vector, with x, y respectively the longitudinal and lateral positions, $\dot{x}, \dot{y}$ the longitudinal and lateral velocities, $\ddot{x}, \ddot{y}$ the longitudinal and lateral accelerations, and α the angle formed between the vehicle center plane and the trajectory. The model is valid under the dynamical constraints of a vehicle, i.e. the maximum values for the longitudinal and lateral accelerations $a_{x,max}, a_{y,max}$ and decelerations $d_{x,max}, d_{y,max}$, and a maximum value for the direction angle α_{max}.

We consider that the ego vehicle navigates with a constant longitudinal acceleration or deceleration profile from its initial velocity v_0, at position x_0, y_0, to a target velocity v_t along the prediction horizon H_P, illustrated in Figure 4.4. The target velocity is bounded by the road's legal speed limits $[v_{min}, v_{max}]$.

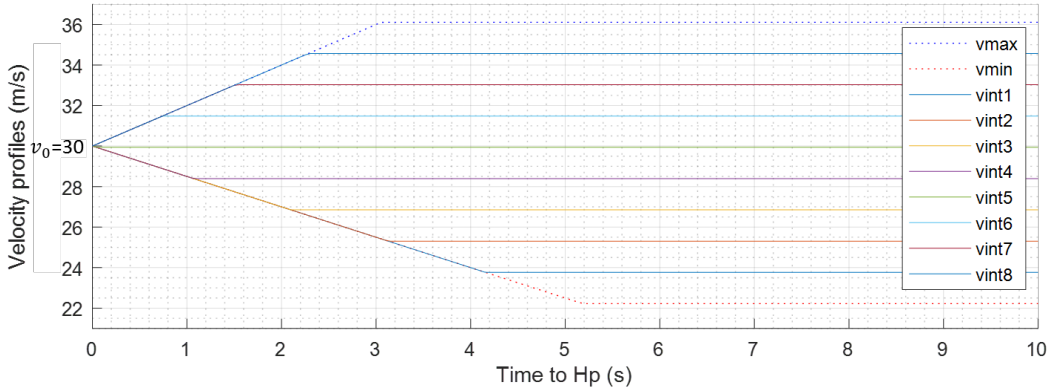


Figure 4.4: Example of 10 intermediate longitudinal velocity profiles with fixed acceleration and deceleration respectively at $a_x = 2\text{m/s}^2$ and $d_x = -1\text{m/s}^2$, bounded velocities $v_{max} = 36.11\text{m/s}$ (130km/h) and $v_{min} = 22.22\text{m/s}$ (80km/h), and the initial ego velocity $v_0 = 30\text{m/s}$ (108km/h).

The vehicle model used in this work is summarized in Equations 4.1a and 4.1b:

$$\begin{cases} \ddot{x}(t) = a_x, \\ \dot{x}(t) = v_x(t) = a_x t + v_0, \\ x(t) = \frac{a_x}{2} t^2 + v_0 t + x_0, \\ \ddot{y}(t) = a_y, \\ \dot{y}(t) = v_y(t), \\ \alpha = \arctan(\frac{v_y}{v_x}), \end{cases} \tag{4.1a}$$

under the constraints

$$\begin{cases} a_x \in [d_{x,max}, a_{x,max}], \\ v_x \in [v_{min}, v_{max}], \\ a_y \in [d_{y,max}, a_{y,max}], \\ \alpha \in [-\alpha_{max}, \alpha_{max}]. \end{cases} \tag{4.1b}$$

The notation v_{ego} is the global ego velocity, expressed as $v_{ego} = \sqrt{v_x^2 + v_y^2}$. As the lateral velocity v_y is usually very small compared to the longitudinal velocity v_x on highways, we approximate $v_{ego} \approx v_x$. Moreover, the condition on the maximum angle α_{max} is also less restrictive than the one on the maximum lateral acceleration $a_{y,max}$ on highways, as given in [Reif & Dietsche 2010]. We thus consider only the constraint on $a_{y,max}$ in the next sections.

4.3.1.3 Trajectory model

The main nominal highway maneuvers are listed in subsection 2.1.3 (p. 18). From this list, we restrict our work to the study of lane keeping, car following, lane changing, passing and overtaking situations. Consequently, the lateral maneuvers are either lane keeping or right/left lane changing:

- The lane keeping maneuver consists in following the center of the current lane: $y(t) = y_{\text{centerline}}$.
- The lane changing maneuver consists in a lane shift from the current lane to the target lane centerline. Based on the literature review in Chapter 2, the sigmoid function has been chosen. Even if the clothoids, lines and circles are the usual shape for highway structure, the mathematical calculation of a clothoid is iterative and leads to a higher calculation time than the one of a sigmoid. Besides, the properties of the sigmoid function make it easy to use, as it is bounded, symmetric, differentiable, and monotonic. We thus approximate the lane change behavior with sigmoids in Equation 4.2, represented in Figure 4.5.

  $$y(t) = y_0 \pm \frac{b}{1 + e^{-\lambda(x(t)-c)}}, \tag{4.2}$$

  with b the shift between y_0 and the asymptotic end value, which corresponds in our problem to the target centerline. The sigmoid parameter λ and the distance delay c are the variables to optimize in our study. They are used for driver safety and comfort,

which depend on the lateral and longitudinal ego vehicle velocities. The maximum slope of the curve is mathematically defined to $b\lambda/4$ at $x(t) = c$, as illustrated in Figure 4.5.

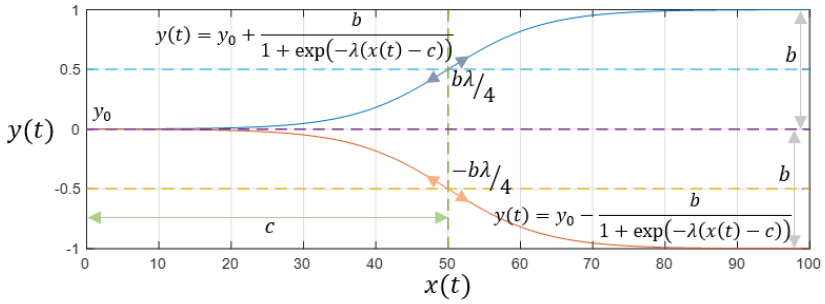


Figure 4.5: Characterization of the sigmoid. For illustration, $x \in [0, 100]$, $b = 1$, $\lambda = 0.15$, and $c = 50$. Both increasing and decreasing sigmoid functions are plotted.

4.3.1.4 Obstacle prediction

We assume constant-speed/direction and full-lane occupancy for any detected obstacle. One of the strategies of the implementation is to replan the reference trajectory if any major change in the surroundings' behavior occurs, e.g. obstacle's lane change, or significant acceleration or deceleration (over 5 km/h). Moreover, lane occupancy on highways is mandatory in the Vienna Convention [United Nations] (Article 11, paragraph 7): "When moving in lines [...], drivers are forbidden, if the lanes are indicated on the carriageway by longitudinal markings, to straddle these markings." In a future work, there is no theoretical limitation for our method to address multiple predicted obstacle's maneuvers with uncertainties. The extension to variable speeds will only impact the longitudinal collision test (see section 4.4.2), whereas the lane change maneuver will result in both the initial and final lanes being marked as occupied.

A front obstacle will thus cover its front space as a collision space. The reminding rear space, i.e. the space behind obstacles will be considered as collision-free for defining the upper bounds of intervals $I_{k=1..K}$. Respectively, a rear obstacle will cover its behind space as a collision space. The front space is then collision-free and use to define the lower bounds of intervals $I_{k=1..K}$. Thereby, an obstacle on the adjacent lane is considered both as potentially a front and a rear obstacle, in case the ego vehicle passes behind or in front of it, as illustrated in Figure 4.6.

4.3.2 Non-Collision Nominal Intervals (NCNI)

The evolution space delimits the spatiotemporal region where the ego vehicle can navigate without collision. This definition considers that whatever happens, the ego vehicle can always apply a kinematically feasible motion within the evolution space without collision. In the scope of this paper, we limit the definition of the evolution space to a nominal highway driving situation by defining the Non-Collision Nominal Intervals (NCNI). According to Figure 4.7, nominal driving conditions consist in a non-conflict situation, i.e, we do not consider the emergency maneuver in a conflict situation.

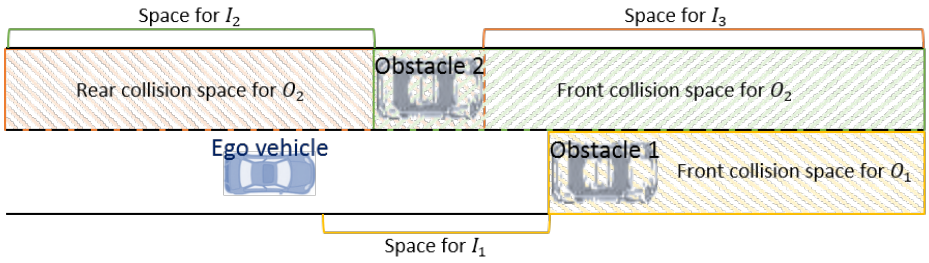


Figure 4.6: Illustration of full lane occupancy obstacles. The striped area shows the front collision space for obstacle O_1 in yellow and obstacle O_2 in green, and the rear collision space for O_2 in orange. The reminding rear space of O_1 defines the space for interval I_1, the reminding rear space of O_2 the space for interval I_2 and the reminding front space of O_2 the space for interval I_3.

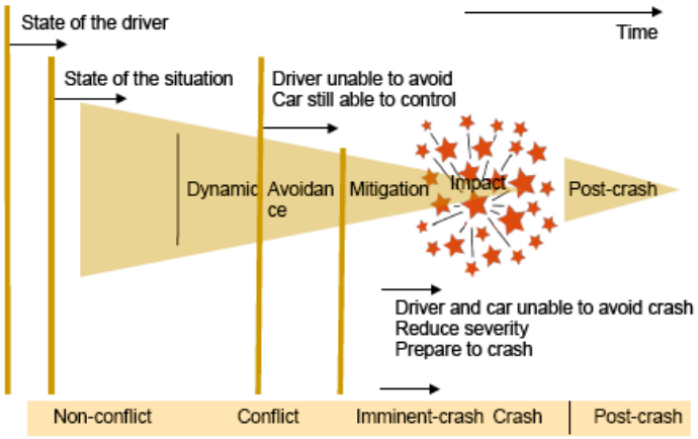


Figure 4.7: The holistic view of safety. Source *Swedish Transport Administration 2010.*

NCNI consist in spatiotemporal intervals, in which the ego vehicle navigates without collision under nominal conditions. We define 3 possible characterizations. First, the ego vehicle passes in front of the obstacle, which is thus considered as a rear obstacle, even if its initial position was in front of the ego vehicle, as obstacle O_3 in Figure 4.8. A rear obstacle yields a lower bound of distance and velocity, under which a collision is encountered. Second, the ego vehicle passes behind the obstacle, which is thus considered as a front obstacle, even if its initial position was behind the ego vehicle, as obstacle O_5 in Figure 4.8. A front obstacle yields an upper bound of distance and velocity, over which there is a collision. Last, if there is no obstacle, an ego phantom vehicle marks respectively the lower and upper bounds, with the minimum and maximum velocity of the ego vehicle. This ego phantom vehicle represents a hypothetic obstacle at the sensors' limit perception range. These bounds delimit the NCNI between the adjacent obstacles, as depicted in Figure 4.8.

The NCNI characterize where, when, and how the ego vehicle can maneuver along the prediction horizon H_P. A classification of the perceived obstacles $O_{j=1..J}$ within the perception area returns the intervals to consider in the ego adjacent lanes. For example in Figure 4.8, the perception area is represented in purple dotted line, 7 obstacles are detected, which form $K = 6$ intervals $I_{k=1..K}$. Each interval I_k is defined by a lower $O_j^{inf_{I_k}}$ and an upper $O_{j'}^{sup_{I_k}}$ bound due to a rear obstacle O_j and a front obstacle $O_{j'}$. In Figure 4.8, I_2 is delimited by

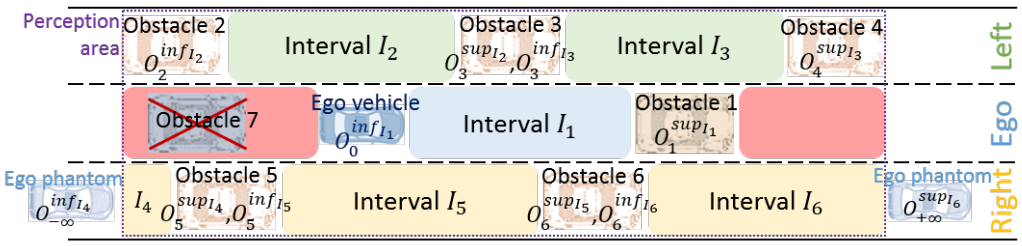


Figure 4.8: Example of a 3-lane road with 7 obstacles around the ego vehicle in the perception area, which define 6 NCNI.

obstacles O_2 and O_3. Thus, I_2 is associated with $(O_2^{inf_{I_2}}, O_3^{sup_{I_2}})$. The open intervals (e.g. intervals I_4 and I_6 in Figure 4.8) are bounded with the minimum and maximum dynamics of the ego phantom vehicle. The ego rear obstacles are not considered as interval bounds, as the ego vehicle is not allowed to reverse on highway in nominal driving conditions. In Figure 4.8, lane/car following will consist in finding the non-collision range of velocities inside interval I_1. A left lane change is only possible if there exists at least one velocity profile to reach intervals I_2 or I_3 from interval I_1, respectively I_4, I_5 or I_6 from interval I_1 for a right lane change.

A spatiotemporal interval is thus defined by a lower and upper bound in distance and velocity. A definition is given in Equation 4.3:

$$I_k \hat{=} \begin{Bmatrix} I_{k,d}(p_i) \\ I_{k,v}(p_i) \end{Bmatrix}_{i=1,N}, \tag{4.3}$$

where p_i is an ego profile, N is the total number of ego profiles, $I_{k,d}$ defines the interval I_k in distance in Equation 4.4a, respectively $I_{k,v}$ in velocity in Equation 4.4b:

$$I_{k,d}(p_i) = [d_{inf}(p_i), d_{sup}(p_i)]_k, \tag{4.4a}$$
$$I_{k,v}(p_i) = [v_{inf}(p_i), v_{sup}(p_i)]_k, \tag{4.4b}$$

where d_{inf}, d_{sup}, are the inferior, superior, distance values, and v_{inf}, v_{sup} are the inferior, superior velocity values to characterize the free space of interval I_k for the ego profile p_i (see subsection 4.4.2 for more details on how to obtain the inf and sup values).

To conclude, the NCNI provide information on longitudinal and lateral position and velocity constraints to avoid a collision.

4.4 Algorithm architecture

This section details the assumptions in subsection 4.4.1 and the algorithm diagrams used to define the NCNI before the SA-search for the reference trajectory in subsection 4.4.2.

4.4.1 Assumptions

The following assumptions are necessary to delimit the nominal use of the proposed motion planner.

Assumption 1 *In the ego lane, only the closest front obstacle is considered, as overtaking this obstacle will lead to a replanning step.*

At least 2 maneuvers are necessary to pass in front of the initial closest front obstacle in the ego lane. However, one of the strategy of our motion planner is to replan if the situation changes significantly, e.g. a lane change or acceleration.

Assumption 2 *If the velocities of the closest front obstacles in each considered left, ego, and right lane within the perception area are below the minimum speed limit, the situation is not nominal and is out of the scope of this motion planner.*

We assign a minimum velocity to navigate on the road in nominal conditions (see subsection 4.3.1.1). If all the closest front obstacles of each lane are below this speed limits, the ego vehicle can not stay without a collision on any of the ego lane, adjacent right or adjacent left, as its minimum velocity is higher than the one of the obstacles, and will lead to a collision. This assumption can be relaxed in some cases, e.g., if the ego vehicle has time and distance for an adjacent lane change first and then to reconsider the situation with the new adjacent lanes.

Assumption 3 *Only the ego, adjacent left and right lanes are considered to define the NCNI. Indeed, if an obstacle in the second left lane plans to navigate to the ego lane, it will necessarily pass through the adjacent left lane first.*

The NCNI are considered only in the ego and adjacent left and right lanes, but further detected obstacles are analyzed for obstacles prediction, specially the ones which plan to navigate on the NCNI's lanes are considered (see subsection 4.3.1.4). There is no theoretical limitation to consider more lanes, except for real-time calculation.

Assumption 4 *If there is a front collision in one of the directions, the corresponding profile is discarded in this direction.*

Assumption 5 *If there is a rear collision in one of the directions all along the prediction horizon, the corresponding profile is discarded in this direction.*

Assumption 6 *Rear obstacles in the ego lane are not considered in a general framework. This assumption can be relaxed in case of specific driver rules, e.g., to pull back the ego vehicle in the mostright lane.*

We will notice in the experimentation Chapter 5 that *Assumptions 4* and *5* are too conservative, and might be relaxed considering an anticipated horizon.

4.4.2 Architecture

The proposed algorithm has three main parts. First, the ego velocity/distance profiles and obstacle profiles are generated, as described in subsections 4.3.1.2 and 4.3.1.4. Second, the intervals are defined and characterized by comparing the ego and obstacles profiles. Last, the remaining collision-free profiles in the sense of the *Assumptions 4* and *5* are gathered through the intervals to define possible maneuvers and associated time-space limitations. The architecture of the algorithm is displayed in Figure 4.9.

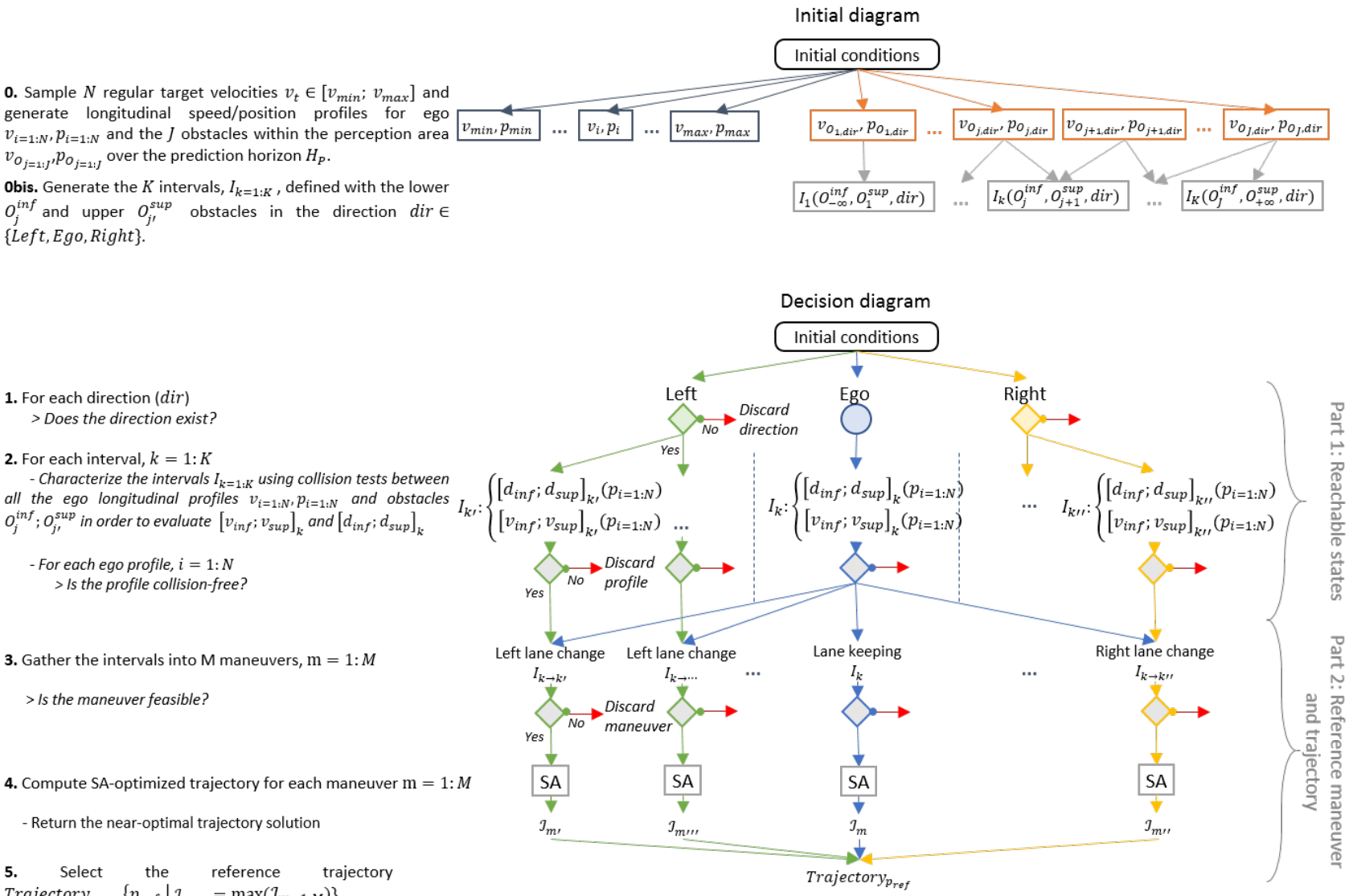


Figure 4.9: Algorithm diagrams.

4.4.2.1 Initial diagram

The initial diagram consists in defining the predicted ego and obstacles motions with the models described respectively in subsections 4.3.1.2 and 4.3.1.4 (step **0**), and generating the Non-Collision Nominal Intervals (NCNI) in step ***0bis***.

Although first and last moments of collision could be evaluated analytically, as it has been done in [Jula *et al.* 2000] using simple vehicle's models, our approach discretizes a sample of N target velocities v_t within the road speed limits $[v_{min}, v_{max}]$. N longitudinal velocity $v_{i=1:N}$ and position $p_{i=1:N}$ profiles are then calculated for the ego vehicle over the prediction horizon H_P. One notices that this discretization step is only used to define the collision limits and the interval bounds, but does not stand for the candidate solution to the reference trajectory, which is calculated in the continuous interval previously defined, see subsection 4.5.1.3.

The longitudinal velocity and position profiles, with indication of the direction, of the J obstacles $O_{j=1:J}$ within the perception area are generated in $v_{O_{j=1..J},dir}$, $p_{O_{j=1..J},dir}$. As detailed in subsection 4.3.2 (p. 91), the space between the obstacles form the NCNI. The intervals I_k are described with their inferior bound based on an obstacle j, O_j^{inf}, their superior bound based on an obstacle j', $O_{j'}^{sup}$, and their direction lane, $dir \in \{Straight, Left, Right\}$: $I_{k=1..K}(O_j^{inf}, O_{j'}^{sup}, dir)$.

4.4.2.2 Decision diagram

The decision diagram treats in parallel each existing direction $left/straight/right$ (step **1**). If the direction does not exist, e.g. there is an end straight lane or no lane to the right or to the left, the direction is discarded. For each interval I_k in the considered direction (step **2**), a longitudinal collision test of Minimum longitudinal Safety Spacings (MSS) [Jula *et al.* 2000] between all the ego longitudinal position profiles and the bounding obstacles of I_k returns the interval characterization $[d_{inf}, d_{sup}]_k$ and $[v_{inf}, v_{sup}]_k$, as depicted in Figure 4.10. MSS returns the authorized traveled distance to maintain safety spacings according to the relative distance and velocity with obstacles. The inf value is the lowest value (resp. velocity and position) from which the ego vehicle is not in collision with $O_j^{inf_{I_k}}$. The sup value is the highest value (resp. velocity and position) up to which the ego vehicle is not in collision with $O_{j'}^{sup_{I_k}}$. For cases with ego phantom, $v_{inf} = v_{min}$, $v_{sup} = v_{max}$, $d_{inf} = x_0$ and $d_{sup} = x(t = H_P, v_{max})$. According to the *Assumptions 4* and *5*, the profiles with collision are discarded for the corresponding direction.

The next step is to gather the intervals into maneuvers (step **3**). A maneuver is either a lane following or a $left/right$ lane change. Each interval of left and right directions is gathered with the straight interval. The maneuver is feasible if there is at least one ego profile which exists in both intervals, i.e., which is not in collision with the lower and upper bound obstacles. For each stored ego profile, the gathering test consists in verifying the condition in Equation 4.5 [Jula *et al.* 2000]:

$$\begin{cases} d_{inf}(left/right) & \leq d_{sup}(straight) \\ d_{inf}(straight) & \leq d_{sup}(left/right) \end{cases} \tag{4.5}$$

If the gathering test is false, the maneuver for the tested ego profile is not feasible and it is discarded. As an illustration in Figure 4.10(d), the left change maneuver $I_{1\to 2}$ is possible

for the ego profile $v_i, v_{i'}$. The lower (resp. upper) velocity bound is the minimum (resp. maximum) of the v_{inf} (resp. v_{sup}) of each existing ego profile: $v_t \in [v_i, v_{i'}]$. The position of the lane shift for each existing ego profile, i.e. the distance delay parameter c of the sigmoid, lies within the minimum and maximum, d_{inf} and d_{sup}, from the initial interval I_k and the target one $I_{k'}$, as expressed in Equation 4.6:

$$c \in [\max(d^{p_n}_{inf_{I_k}}, d^{p_n}_{inf_{I_{k'}}}); \min(d^{p_n}_{sup_{I_k}}, d^{p_n}_{sup_{I_{k'}}})] \tag{4.6}$$

as illustrated in Figure 4.10d with I_1 and I_2 for ego profiles v_i, p_i in blue and $v_{i'}, p_{i'}$ in purple, i.e. $n = i, i'$.

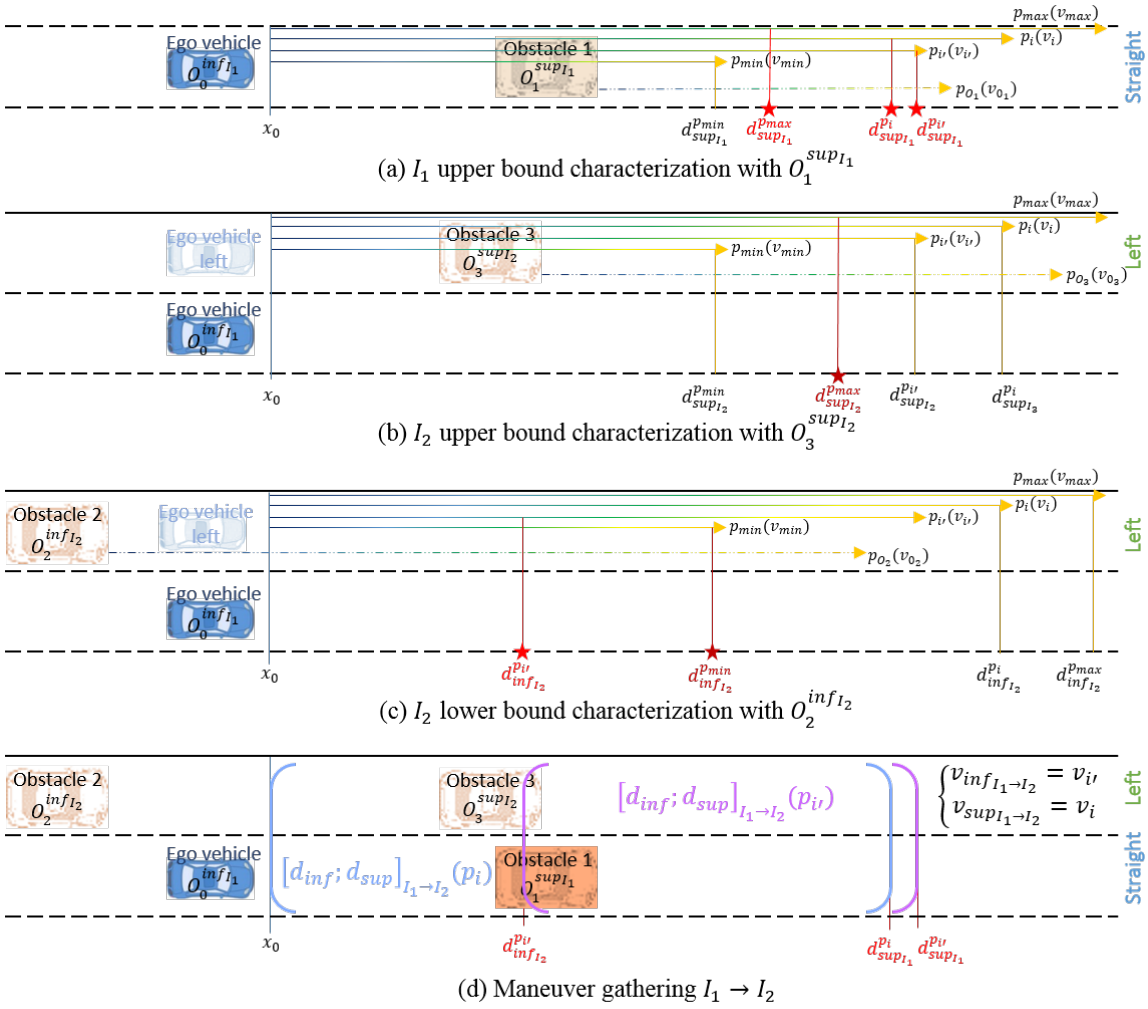
Then, SA optimization, using the evaluation function of Chapter 3 ($\mathcal{J}_{obj,m} = \mathcal{I}_m$), is applied (step **4**) in order to return a near-optimal trajectory solution for each remaining maneuver m. The final step **5** selects the near optimal trajectory solution which has the best evaluation function values $\mathcal{I}_m$. This trajectory is the reference trajectory sent to the controller. Next section details the SA algorithm.

4.5 Simulated Annealing optimization

As stated previously, the meta-heuristics algorithms represent a good solution to problems with black-box objective functions, huge number of potential solutions, and acceptance of a near-optimal solution. Meta-heuristics for optimization aim at minimizing or maximizing an objective function by selecting the best candidate solution in the search space. Among the variety of meta-heuristics, the Simulated Annealing (SA) algorithm presents the following advantages which are useful for our problem: (i) it does not need to compute any derivative of the objective function, i.e., it can be applied to non-differentiable functions and non-explicit ones; (ii) it can avoid local minima by giving the possibility to accept worst solutions; (iii) the computation time is tunable with a trade-off on the optimality; and (iv) it considers the variables to optimize as a single candidate solution, i.e. the variables can be dependent from each other. The main drawback is the fine experimental tuning required in order to guarantee a good behavior of the algorithm. This section describes the algorithm in subsection 4.5.1 and its evaluation in subsection 4.5.2. In the following, the objective function is called the evaluation function when it references to the output of Chapter 3.

4.5.1 Algorithm description

The SA is based on an analogy with the annealing technique used in metallurgy to minimize the energy of a material by slowly cooling it with irregular heating steps. This strategy allows the material to reorganize its crystals and to find its most stable configuration with the minimum possible energy. This methodology has been developed independently by [Kirkpatrick *et al.* 1983] and [Černỳ 1985]. The main idea of the algorithm is thus to accept a worse solution according to an acceptance probability, slowly decreasing. In other words, the notion of slow cooling in the physics approach is interpreted in the SA algorithm as a slow decrease in the probability of accepting worse solutions. Thus, the heating step is the acceptance of this worse solution. This phenomenon is introduced by a global time-varying temperature parameter. The higher the temperature is, the more worse solutions are accepted (bigger



(a) I_1 upper bound characterization with $O_1^{sup_{I_1}}$

(b) I_2 upper bound characterization with $O_3^{sup_{I_2}}$

(c) I_2 lower bound characterization with $O_2^{inf_{I_2}}$

(d) Maneuver gathering $I_1 \rightarrow I_2$

Figure 4.10: Characterization of the intervals I_1 and I_2 for a left change maneuver. The collisions are depicted in red, the eliminations in dark red. (a) Ego I_1 upper bound: The ego profiles $p_{max}, p_i, p_{i'}$ are in collision with $O_1^{sup_{I_1}}$, they are discarded for the direction *ego* (*Assumption 4*). (b) Left I_2 upper bound: The p_{max} profile collides with $O_3^{sup_{I_2}}$. (c) Left I_2 lower bound: Both $p_{min}, p_{i'}$ are in collision with $O_2^{inf_{I_2}}$. p_{min} always collides along H_P, so the profile is discarded (*Assumption 5*). (d) Left change maneuver $I_{1\rightarrow 2}$: As p_{max} collides both with I_1 and I_2, it is discarded. Only $p_{i,i'}$ remain with $[d_{inf}; d_{sup}]_{I_{1\rightarrow 2}}(p_i)$ and $[d_{inf}; d_{sup}]_{I_{1\rightarrow 2}}(p_{i'})$.

probability). The lower the temperature is, the less worse solutions are accepted (smaller probability). With a too low, respectively too high, temperature value, the SA becomes like stochastic hill climbing, respectively random search. Besides, the positive aspect in accepting worse solution is a more extensive search for the global optimal solution, as well as the ability to jump out of local optima.

The SA works as follows: (0) selects an initial candidate solution and fixes an initial temperature; (1) selects a random solution close to the current one; (2) evaluates this solution and decides to accept or reject it if either it is a better local optimum to the objective function, or it has a non-zero probability of acceptance as a worse solution; (4) decreases the temperature parameter to decrease the probability of acceptance for a worse solution.

The drawback of SA is the need to be empirically adjusted for each problem. The tuning of the annealing schedule concerns both the resolution scheme and the definition of the parameters. More information on the different areas of tuning can be found in [Aarts & Van Laarhoven 1985]. We detail the adaptation of the SA to our problem considering the architecture choice in subsection 4.5.1.1, the cooling parameters in subsection 4.5.1.2, and the variables search space in subsection 4.5.1.3.

4.5.1.1 Architecture choice

Authors in [Eglese 1990] review different improvements to the SA architecture. The main difference concerns the optimization loop based on a *step-temperature* with the pseudo-code in Algorithm 1 or a *continuous-temperature* with the pseudo-code in Algorithm 2. The *step-temperature* consists of two optimization loops: the temperature decreases in steps in an outer loop, and at each temperature step, an inner loop is applied to find a local equilibrium solution. The *continuous-temperature* consists in one optimization loop which corresponds to the outer loop of the first method, i.e., the temperature decreases continuously at each iteration. The first architecture is closer to the physical law but the second one is faster. In this work, we use the second one for real-time application. The corresponding algorithm is written in Algorithm 3. Moreover, we list the best encountered solution, which corresponds to the global near-optimal solution found until then.

4.5.1.2 Parameters definition

The annealing schedule includes the tuning of:

- Acceptance probability function: We use the initial Metropolis rules with the Boltzmann probabilities to randomly accept a worse solution: $\mathcal{P}(T) = e^{(-\Delta(\mathcal{J}_{obj})/T)}$.

- Initial temperature T_0: The temperature T is a positive global-time varying parameter used for the acceptance probability function $\mathcal{P}(T)$, which decreases along iterations. The higher T is, the more uphill candidates can be accepted as a solution given the acceptance probability function $\mathcal{P}(T)$, and vice versa. T_0 can be calculated by fixing the initial acceptance probability: with Z a characteristic value of the objective function $\mathcal{J}_{obj}$ (e.g. median or maximum gap), for a given initial acceptance probability P_0, the initial temperature is obtained from $P_0 = e^{(-Z_0/T_0)}$.

Algorithm 1: SA pseudo-code for the *step-temperature* architecture.

```
Select an initial state i ∈ S;
Select an initial temperature T_0 > 0;
Set counter τ = 0;
repeat
    Set repetition counter n = 0;
    repeat
        Generate state j, a neighbor of i;
        Calculate Δ = J_obj(j) − J_obj(i);
        if Δ < 0 then
            i := j;
        else if rand(0,1) < exp(−Δ/T) then
            i := j;
        n = n + 1;
    until n = N(τ)
    τ = τ + 1;
    T = T(τ);
until stopping criterion true
```

Algorithm 2: SA pseudo-code for the *continuous-temperature* architecture.

```
Select an initial state i ∈ S;
Select an initial temperature T_0 > 0;
Set counter τ = 0;
repeat
    Generate state j, a neighbor of i;
    Calculate Δ = J_obj(j) − J_obj(i);
    if Δ < 0 then
        i := j;
    else if rand(0,1) < exp(−Δ/T) then
        i := j;
    τ = τ + 1;
    T = T(τ);
until stopping criterion true
```

- Cooling schedule: The cooling schedule is the iterative function to decrease the temperature value. Different strategies exist such as linear, geometric, logarithmic or adaptive cooling schedules. The convergence is guaranteed if the temperature decreases in a logarithmic way [Hajek 1988]. However, in practice, the geometric progression, $T_n = q \cdot T_{n-1}$, is faster and returns satisfying results. The geometric parameter q is calculated according to the final acceptance probability $\mathcal{P}_f$ or temperature value T_f: $\mathcal{P}_f = e^{(-Z_f/T_f)}$, $T_f = q^N \cdot T_0$, with Z_f a characteristic value at the end of the N iteration process.

- Initial solution ξ_0: The initial solution is the current trajectory of the ego vehicle,

Algorithm 3: SA pseudo-code used in our work.

```
Select an initial state i ∈ S;
Set the optimal encountered state e := i;
Select an initial temperature T_0 > 0;
Set counter τ = 0;
repeat
    Generate state j, a neighbor of i;
    Calculate Δ = J_obj(j) − J_obj(i);
    if Δ < 0 then
        i := j;
    else if rand(0,1) < exp(−Δ/T) then
        i := j;
    if J_obj(j) < J_obj(e) then
        e := i;
    τ = τ + 1;
    T = T(τ);
until stopping criterion true
```

respecting Equation 4.11 (p. 102): $v_t = min(max(v_0, v_{min}), v_{max})$, $c = (d_{inf} + d_{sup})/2$, $\lambda = 4/c$.

- Neighborhood search: The function generates uniformly random numbers for the creation of new candidate solutions. The main point is to define the length of the interval for the neighborhood search. Indeed, if the search is made in the entire space, the reasoning is the one of random search. If the search space is too small, the algorithm will frequently get trapped in a local minimum too quickly. Moreover, if the length of the interval is fixed, the methods is close to a local random search. In order to maintain the idea of the annealing process, the length of the search interval is converted in a decreasing function. We choose arbitrarily a proportionally decreasing interval length with the temperature: $(ub - lb)_{iter} = T_{iter}/T_0 \cdot (ub - lb)_0$.

- Stop criterion: The stop criterion is either fixed as a maximum number of iterations, or when the temperature is below a predefined T_f, or at a steady state of the evolution function, i.e. $\Delta(\mathcal{J}_{obj}) \leq \varepsilon$.

4.5.1.3 Variables search space

The variables to optimize for each possible maneuver m are the target velocity v_t, the sigmoid parameter λ and the distance delay c (see Equation 4.2 (p. 90)). We note ξ the variables vector, i.e. $\xi = (v_t, c, \lambda)$. The intervals characterization gives the bounds of the variables search space, but the search is continuous inside the intervals.

The variable v_t is bounded in Equation 4.11 by the velocity intervals of the non-discarded ego profiles calculated in subsection 4.4.2.2 (p. 96). For a lane change maneuver, c is bounded by the position intervals in Equation 4.11 calculated in Equation 4.6 (p. 97). The position interval depends on v_t, thus it is interpolated from the position intervals of the remaining

speed profiles. The bounds of λ are issued from the constraint of a lane change maneuver completion, see Figure 4.11, and one of the maximum lateral accelerations of the sigmoid function to respect the vehicle's dynamics limits, see Equation 4.2 (p. 90). The first constraint is defined by requiring the sigmoid curve completion, for example at 98% [Cesari *et al.* 2017]: $0.98b \leq y(x = 2c) \iff \lambda \geq \frac{4}{c}$. The latter constraint is given by the maximum value of the second-order time derivative (with $y = y(t) - y_0$), which respects the limitation value for the ego lateral acceleration $a_{y,max}$. The detailed calculation is written in Annexe A.1 (p. 165). The results are given in Equations 4.7 and 4.8:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = a_x \frac{\mathrm{d}y}{\mathrm{d}x} + v_x^2 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} \tag{4.7}$$

$$= \frac{a_x \lambda}{b} y(b-y) + \frac{v_x^2 \lambda^2}{b^2} y(b-y)(b-2y) \tag{4.8}$$

$$\leq a_{y,max} \tag{4.9}$$

Equation 4.7 is obtained according to Schwarz's theorem and chain rule formula for computing the derivative of the composition of two or more functions with continuous second-order derivatives functions. Equation 4.8 uses the derivative properties of the sigmoid function.

The two terms in Equation 4.8 are bounded separately, so that the maximum value for λ respecting Equation 4.9 is approximated by Equation 4.10. More details are written in Annexe A.2 (p. 167).

$$\lambda_{max} = \frac{-a_x b/4 + \sqrt{\Delta}}{\frac{\sqrt{3}}{9} v_x^2 b}, \Delta = \frac{a_x^2 b^2}{16} + \frac{2\sqrt{3}}{9} v_x^2 b a_{y,max} \tag{4.10}$$

Thus, the variables search space of ξ for $I_{k \to k'}$ is illustrated in Figure 4.12 and defined in Equation 4.11:

$$\begin{cases} v_t \in [v_{inf}, v_{sup}]_{I_{k \to k'}} \\ c \in [d_{inf,v_t}, d_{sup,v_t}]_{I_{k \to k'}} \\ \lambda \in [\frac{4}{c}, \lambda_{max,v_t}] \end{cases} \tag{4.11}$$

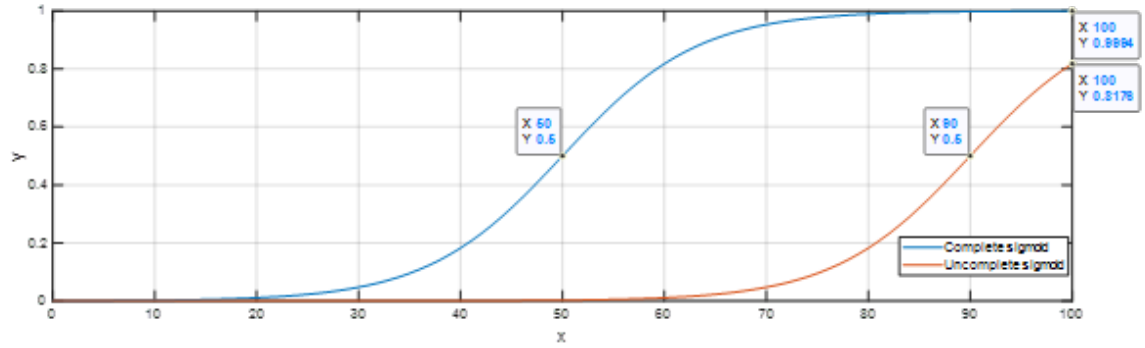


Figure 4.11: Illustration of the sigmoid completion constraint. For $\lambda = 0.15$, $x = [0, 100]$, the sigmoid in blue with $c = 50$ is complete at 99.4%, whereas the one in orange with $c = 90$ is incomplete at 81.76%.

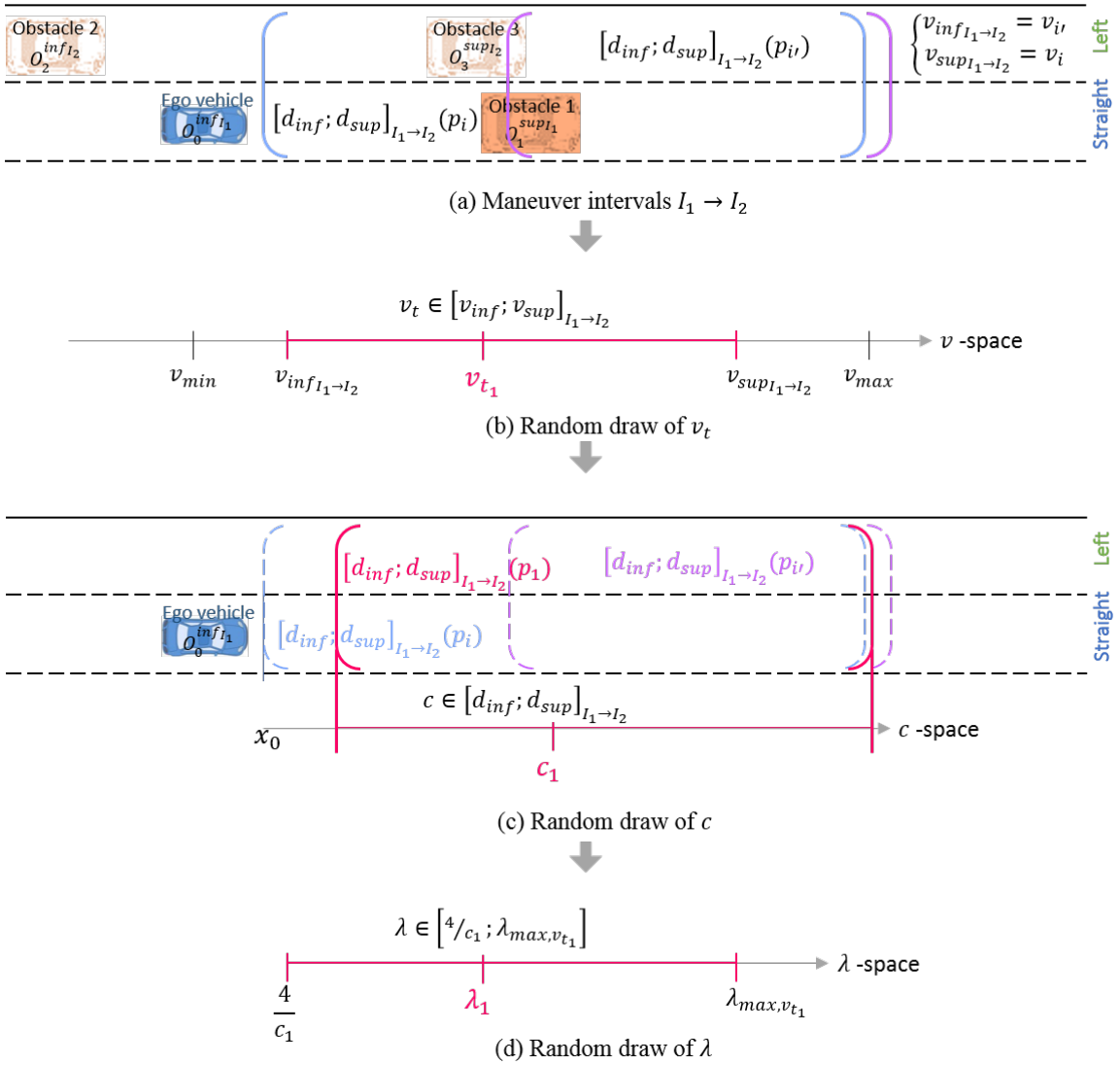


Figure 4.12: Illustration of the variables search space. The maneuver interval in (a) returns the velocity space in (b) to select the target velocity v_{t_1}. The selected target velocity v_{t_1} is used to define the c-space interval in (c). The selected c_{t_1} is used to define the λ-space interval in (d), in which λ_{t_1} is selected.

4.5.2 Evaluation

We evaluate the performance of the algorithm by analyzing the convergence to a near-global optimum and the sensitivity to T_0 and q, as well as to the initial variables $\xi_0 = (v_t, c, \lambda)_0$. The algorithm is run 10 times for each performance test. The evaluation tests are performed for a representative case study of a left lane change maneuver with one obstacle in the ego front lane, with a relative velocity $v_{rel,O_1} = -5$ m/s and a relative distance $d_{rel,O_1} = 90$ m. The

ego initial velocity is $v_0 = 30$ m/s, and 15 v_t sampled within $[v_{min}$=22.22, v_{max}=36.11] m/s are chosen.

4.5.2.1 Objective function

We want to maximize the risk assessment function previously developed in Chapter 3. It is based on the Fuzzy Dempster-Shafer Theory and is not explicit. We do not have any a priori on this function. For the evaluation of the SA, we selected the three risk's criteria of category 3 (Obstacle safety in Table 3.1, p.55): the ego velocity v_{ego}, the relative velocity v_{rel,O_j} and the time headway T_{h,O_j} for each obstacle within the perception area ($O_{j=1..J}$) expressed with the predictive operators in Equations (4.12a), (4.12b) and (4.12c):

$$v_{rel,O_j} = mean(v_{ego} - v_{O_j}), \tag{4.12a}$$

$$T_{h,O_j} = min(\frac{x_{O_j} - x_{ego}}{v_{ego}}), \tag{4.12b}$$

$$v_{ego} = end(v_x) = v_x(t = H_p) \tag{4.12c}$$

Figure 4.13 displays the evaluation of this black-box objective function. The choice of v_t influences all 3 criteria, while λ and c have an effect on T_h. One notices that the maximum of the evaluation function in yellow leads to a higher time headway, a negative relative velocity (i.e. the ego vehicle is navigating faster), and an ego velocity inside the speed limits of the road. Besides, the discontinuities from dark blue to green evaluation function at $T_h \approx 2$ s is due to the rule-base in Table 3.2. Indeed, the warning levels to pass from T_h in unsafe distance to T_h in safe distance for the same relative velocity and ego velocity membership are not smooth. For example, if relative velocity belongs to negVelocity and ego velocity to fastVelocity, the warning risk level, which results from the rule-base, is denoted as very high for unsafe time headway (rule number 39) and low for safe time headway (rule number 42).

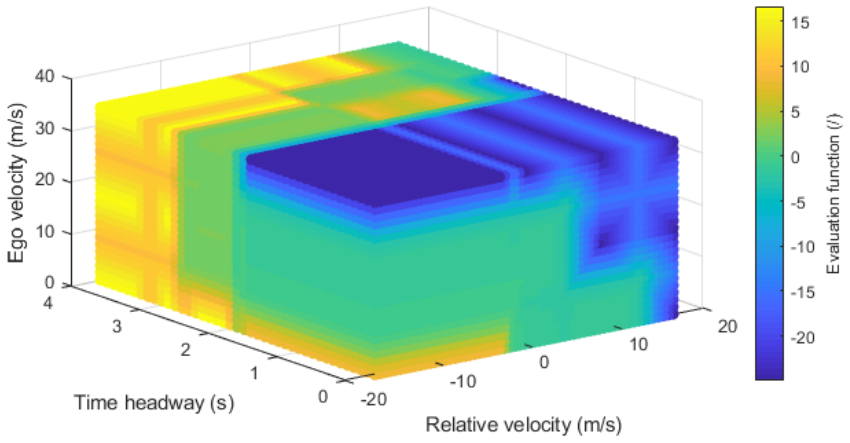


Figure 4.13: Evaluation function over the range of 3 variables: relative velocity v_{rel}, time headway T_h and ego longitudinal velocity v_x.

4.5.2.2 Convergence

The convergence analysis yields the convergence speed and optimal evaluation value $\mathcal{I}_m$ displayed in Table 4.1. One notices that the convergence speed is fast, as the evaluation function displayed in Figure 4.13 shows similar values for large ranges of variables. This is due to a conservative evaluation for decision making. Moreover, the optimal values over the 10 runs present a very low standard deviation σ, which indicates that despite the heuristic calculation, the algorithm always finds a solution close to the global optimum value.

Table 4.1: SA convergence evaluation.

		mean	*min*	*max*	σ
Convergence	Speed (iteration)	112	90	135	/
	Optimal $\mathcal{I}_m$ (/)	5.99	5.98	5.99	$7.4e^{-3}$

4.5.2.3 Temperature sensitivity

In order to verify the annealing schedule tuning, we test 7 values for $T_0 \in \{10^{-3}; 10^{-2}; 10^{-1}; 10^{0}; 10^{1}; 10^{2}; 10^{3}\}$ and 10 values for q sampled within $[0.8, 0.99]$. The averaged maximum evaluation function is plotted in Figure 4.14. The maximum of the evaluation function is reached for $T_0 \in [0.001, 10]$ with $q \in [0.90, 0.99]$.

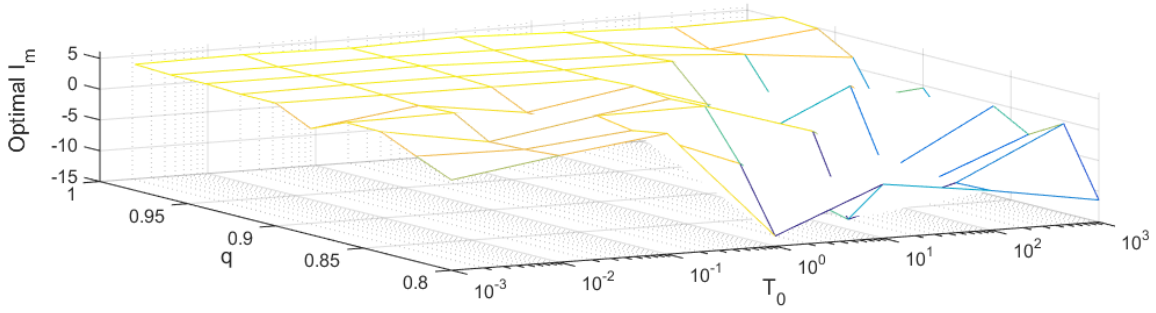


Figure 4.14: Influence of T_0 and q on the maximum evaluation value.

4.5.2.4 Initial variables sensitivity

To guarantee a uniform random search, 10 different initial variables sets $\xi_{0,1..10}$ are tested. The results are summarized in Table 4.2. The σ of the optimal $\mathcal{H}_m$ indicates that the initial variables choice has a limited influence on the optimization process. However, as the evaluation function shows similar results for large ranges of variables, the optimal parameters have high standard deviation.

Table 4.2: SA performance evaluation.

		mean	*min*	*max*	σ
Variables Initial Values	Optimal v_t (m/s)	23.40	22.22	28.07	2.47
	Optimal c (m)	136	98	193	32
	Optimal λ (m^{-1})	0.068	0.029	0.102	0.022
	Optimal $\mathcal{I}_m$ (/)	5.65	4.67	5.99	0.41

4.6 Numerical example

We demonstrate our SA-optimized trajectory generator on the scenario of Figure 4.8 (p. 93). We detail the scenario values in subsection 4.6.1, the SA parameters in subsection 4.6.2, and the maneuvers evaluation in subsection 4.6.3.

Moreover, to validate our approach in real-time, 28 computations of the described scenario are performed on an embedded computer (2.10 GHz Intel Core i7-3612QM CPU, 8 GB RAM) running a Visual C/C++ Solution File for Matlab/Simulink Coder. The mean ± standard deviation/minimum/maximum values for the calculation time are $72 \pm 10/61/99$ ms, which is satisfyingly fast for a real-time predictive motion planner.

4.6.1 Scenario description

We assign for each obstacle $O_{j=1..6}$ the following initial relative velocity (in m/s), position (in m) and direction: $O_1(-5, 90, ego)$, $O_2(0, -30, left)$, $O_3(0, 60, left)$, $O_4(5, 130, left)$, $O_5(-8, -20, right)$, $O_6(-5, 70, right)$, summarized in Figure 4.15. The lateral relative position is defined as the entire lane (see section 4.3.1.4): *ego*, *left* or *right*.

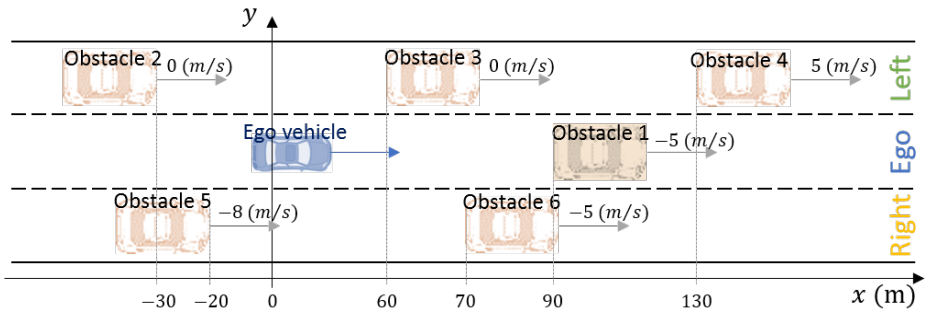


Figure 4.15: Illustration of the numerical scenario.

The ego vehicle's initial position is $x_0 = 0$ m, $y_0 = y_{\text{centerline-ego}}$, with $v_0 = 30$ m/s, $a_x = 2$ m/s^2 and $d_x = -1.5$ m/s^2. According to [Reif & Dietsche 2010], the limitation value for the lateral acceleration on highways in nominal conditions (i.e. comfort) is $|a_{y,max}| = 2$ m/s^2. We sample 15 target velocities v_t for the ego vehicle within $[v_{min}$=22.22, v_{max}=36.11] m/s. The prediction horizon H_P is set to 10 s.

4.6.2 SA parameters

We use the black-box decision function in Chapter 3 with the 3 criteria (relative velocity v_{rel}, time headway T_h and ego velocity v_x) recalled in section 4.5.2.1 and displayed in Figure 4.13. The range of the criteria are $v_{rel} \in [-8, 14]$ m/s, $T_h \in [0, 3.36]$ s, and $v_x \in [22.22, 30.16]$ m/s. A discretized zoom of the evaluation function is depicted in Figure 4.16. One should notice that all the combination of the ranges of criteria values are not feasible, which implies that the maximum value of the evaluation function depicted in Figure 4.16 is not necessary reached during the simulation.

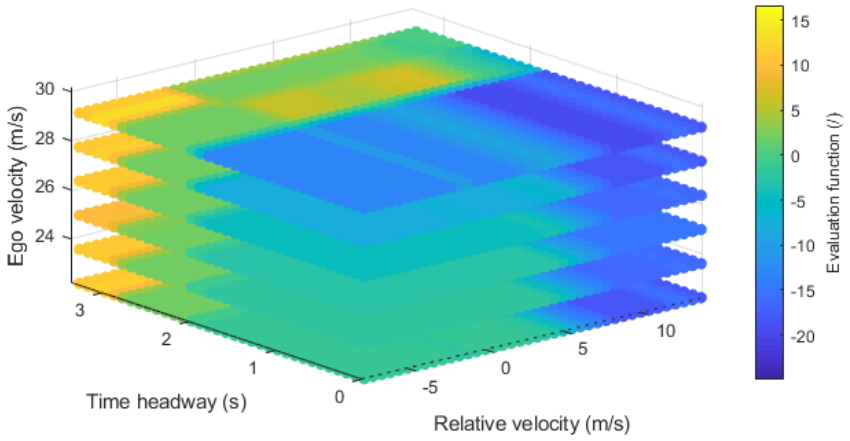


Figure 4.16: Representation of the black-box evaluation function $\mathcal{J}_{obj}$.

In order to propose a real-time algorithm, we set a maximum of 200 iterations and a steady state criterion of 10^{-1} on the evaluation function $\mathcal{J}_{obj}$. We consider initial and final acceptance probability $P_0 = 0.2$ and $P_f = 0.001$, and initial and final characteristic value of the evaluation function $Z_0 = 1$ and $Z_f = 0.1$. Thus the initial temperature is $T_0 = 0.62$ and the geometric parameter is $q = 0.98$.

4.6.3 Maneuvers evaluation

The maneuvers to study are:

- Lane following: to stay in interval 1, $I_{1\to1}$;
- Left lane change: to pass from interval 1 to interval 2, $I_{1\to2}$, or from interval 1 to interval 3, $I_{1\to3}$;
- Right lane change: to pass from interval 1 to interval 4, $I_{1\to4}$, or from interval 1 to interval 5, $I_{1\to5}$, or from interval 1 to interval 6, $I_{1\to6}$.

In this case study, $I_{1\to3}$ does not allow a left change maneuver, as O_3 is driving at the initial ego velocity, and thus, the distance to reach a higher v_t is too low with O_1 for a lane change within the prediction horizon. There is no maneuver for $I_{1\to4}$ neither, as O_4, which is behind the ego vehicle at the initialization, is driving at the minimum bound velocity of the ego vehicle. No maneuver are allowed for $I_{1\to6}$ as O_1 and O_6 have the same velocity and

are close to each other, Equation 4.5 is not respected for any of the velocity profile. Only maneuvers $I_{1\to1}$, $I_{1\to2}$ and $I_{1\to5}$ remains.

The index profiles and velocity bounds for each maneuver, as well as the optimal parameters and evaluation, are summarized in Table 4.3. The corresponding $\mathcal{I}_m$ lies in $[-0.42, 6.00]$. The optimization process is thus to find a trajectory with the maximum value of $\mathcal{I}_m = 6$.

Table 4.3: Maneuvers bounds and evaluation.

Maneuver	$I_{1\to1}$	$I_{1\to2}$	$I_{1\to3}$	$I_{1\to4}$	$I_{1\to5}$	$I_{1\to6}$
N	1..7	9	none	none	2..5	none
v_{inf} (m/s)	22.22	30.16	none	none	23.21	none
v_{sup} (m/s)	28.17	30.16	none	none	26.19	none
v_t (m/s)	28.17	30.16	none	none	26.19	none
c (m)	n.a.	116	none	none	157.5	none
λ (m^{-1})	n.a.	0.034	none	none	0.025	none
$\mathcal{I}_m$ (/)	6.00	0.07	none	none	3.67	none

- $I_{1\to1}$: The target velocity profiles 1 to 7 are valid for staying in the ego lane behind obstacle O_1 without collision. The maneuver is thus to break, which is logic as O_1 has a smaller velocity than the initial ego one set to 30 m/s. The minimum allowed target velocity is 22.22 m/s and the maximum one is 28.17 m/s. With the evaluation function $\mathcal{J}_{obj}$ in Figure 4.16, one notices that a relative velocity of ±5 m/s has no influence. Moreover, Th is determined with the minimum value along the prediction horizon. In fact, $p = 1..7$ are constant and decreasing velocity profiles, thus their Th are the same and equal to Th at the beginning of the simulation. Thereby, only the ego velocity modifies the evaluation function, which is higher with higher ego velocity. Finally, the best target velocity is 28.17 m/s.
- $I_{1\to2}$: Maneuver $I_{1\to2}$ is only possible with $p = 9$ velocity profile at 30.16 m/s. It has a very low evaluation value due to the small T_h to insert between O_2 and O_3.
- $I_{1\to5}$: The target velocities 2 to 5 are allowed for $I_{1\to5}$, i.e., the target velocity is in $[23.21, 26.19]$ m/s. As for $I_{1\to1}$, the higher ego velocity is chosen: $v_t = 26.19$ m/s. It has a smaller evaluation value than $I_{1\to1}$ as the distances to O_5 and O_6, as well as v_t, are smaller than the ones with O_1, and v_{rel} is higher. But it has a better evaluation value than $I_{1\to2}$, as O_5 has a smaller velocity and thus a bigger time headway than the left lane change with O_2.

Finally, the best maneuver with the maximum I_m value is to stay in the ego lane $I_{1\to1}$ and adapt the ego velocity for car following at $v_t = 28.17$ m/s.

4.7 Conclusion and future work

This chapter presents an all-in-one architecture for trajectory generation and decision making, based on a black-box objective function. The method addresses both the problem of

candidate trajectories discretization and path/velocity decoupling optimization using first a velocity representation for the evolution space. It is then reduced to the complement of the ICS in nominal situations by introducing the Non-Collision Nominal Intervals (NCNI). A SA-optimized sigmoid trajectory within NCNI is finally performed to define a near-optimal reference trajectory. Moreover, this optimization strategy can run in real-time for black-box objective functions. Besides, if there is one or more intervals allowing the definition of a maneuver, it is guaranteed by the collision test that the maneuver is safe according to the minimal safety spacings, i.e. with no collision under the nominal driving conditions. Thus, even if the algorithm does not converge to the optimal or has no time to converge, the returned path is safe. However, the major drawback of this algorithm is the need of obstacles profiles over a prediction horizon.

Future work will consist in considering more complex speed profiles with the addition of the acceleration profile as a fourth optimization variable, and extending to more complex behaviors of obstacles with lane changing, acceleration and deceleration, as well as merging, with a more aggressive decision function. Lastly, some assumptions could be relaxed for a better adaption to real driving conditions.

The proposed architecture, as well as the Fuzzy Dempster-Shafer (FDS) decision making and the Simulated Annealing (SA) optimized trajectory generation, are implemented and validated in real experiments in the next chapter.

Chapter 5

Experimentation and Results

Contents

This chapter presents the results of the experimental tests performed to validate the motion planning approach proposed in this thesis. The experimental platform is the prototype vehicle of Institut VEDECOM navigating on the Versailles-Satory test tracks, France. The validation considers for different scenarios, from a simple lane following to a double lane changing with one or two obstacles. All the scenarios had been previously tested in simulation. Our solution generates a trajectory for the studied situations of lane following, car following, lane changing and overtaking with dynamic obstacles, without any prior knowledge. The developed algorithm has been integrated with localization, communication, obstacles perception, and control modules.

This chapter is structured as follows: First, section 5.1 describes the experimental environment used to test our motion planning algorithm. The tested scenarios with results are presented in section 5.2. The conclusions of the performed experiments are summarized in section 5.3.

5.1 Experimental environment

This section describes the experimental environment and architecture, respectively in subsections 5.1.1 and 5.1.2, used to test and implement our motion planning algorithm.

5.1.1 Environment description

The experiments have been performed on the test tracks described in subsection 5.1.1.1, with the prototype vehicle of Institut VEDECOM, which is presented in subsection 5.1.1.2. This vehicle is equipped with different sensors to locate itself on the road and to detect obstacles as presented in subsection 5.1.1.3. Finally, in order to gather the appropriate information, we worked with three reference frames defined in subsection 5.1.1.4.

5.1.1.1 Test tracks

The experiments have been conducted on the test tracks of Versailles-Satory (78000), France. In order to navigate in an environment similar to highways, we selected the high speed test track showed in Figure 5.1.

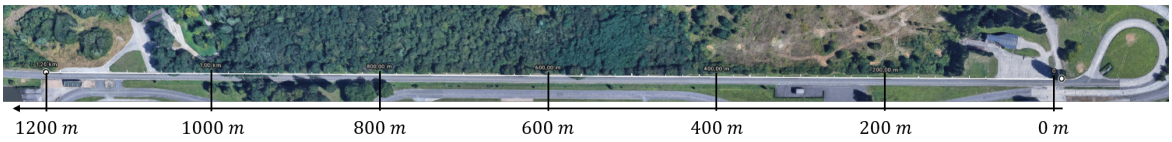


Figure 5.1: Versailles-Satory high speed tracks, 78000 Versailles, France. [credits: GoogleMaps]

The road input for our algorithm is modeled offline as a 2-lane straight road, 1.2 km long, with a constant lane width equal to 3.7 m and constant road and speed limits.

5.1.1.2 Ego vehicle prototype

The experimental autonomous vehicle of Institut VEDECOM is an electric Renault Zoe; see Figure 5.2. It has full drive-by-wire capabilities, with an autonomous mode allowed from 5 km/h to 50 km/h. Thus, the velocity of the vehicles during our experiments is in the range $[20, 45]$ km/h to respect the technological constraints of the prototype vehicle. Even if the experimental velocity range is below the highway velocity range (e.g $[60, 130]$ km/h in France), our experiments demonstrate and validate the efficiency and reliability of our algorithm in different scenarios. Moreover, the simulation sections for decision making 3.4 (p. 74) and trajectory generation 4.6 (p. 106) show the good behavior of our method in high velocity environments.

In order to perceive its own motion and the surroundings, the ego vehicle is equipped with different sensors, such as camera, LIDAR, radar, GPS, Inertial Measurement Unit (IMU), etc. We only detail the sensors used in our experiments:

- IXBLUE ATLANS-C Mobile Mapping INS/GNSS RTK (Septentrino)/odometer (Correvit) to locate the ego vehicle in the world frame with a centimeter level accuracy;
- 5 LIDAR Ibeo LUX to detect obstacles;

(a) Outside view.

(b) Inside view

Figure 5.2: Renault Zoe, experimental test vehicle of Institut VEDECOM.

- Ultra High Frequency (UHF) communication module to receive localization data from other vehicles.

The algorithms run on an embedded computer (2.10 GHz Intel Core i7-3612QM CPU 8 GB RAM).

5.1.1.3 Obstacles vehicles

The test scenarios involve either one or two mobile obstacles. These vehicles are manually driven. The first one, an electric Renault Zoe, has the same localization solution as the ego vehicle, IXBLUE ATLANS-C, and a UHF communication transmitter to send its state vector $\chi_{O_1} = (x, y, v)$ (longitudinal position, lateral position, velocity) in the world frame $\mathscr{W}$ to the ego vehicle. The second one, a Renault Clio, is not instrumented and was localized by a perception algorithm based on the LIDAR of the ego vehicle. The perception algorithm is detailed in subsection 5.1.2.2. It returns the obstacle state vector $\chi_{O_2} = (x, y, v)$ (longitudinal position, lateral position, velocity) in the ego frame $\mathscr{E}$.

The nominal behavior of the obstacles is to keep their velocity approximately constant with a straight direction. In order to show real situation, we ask the obstacles to change their velocity or direction along the scenarios. However, no information has been set up a priori in the algorithm for these tests. Besides, the perception algorithm does not return any uncertainty information and the localization module ATLANS-C has a certainty close to 100%; the uncertainty approach of our algorithm could thus not be tested.

5.1.1.4 Reference frames

We use the following four frames, see Figure 5.3 for an illustration:

- World frame $\mathscr{W}$: Geographic coordinate system defined by the latitude, longitude and orientation to the East. The (latitude, longitude) coordinates are then transformed into a 2-D frame $(x, y)_{\mathscr{W}}$.

- Road frame $\mathscr{R}$: As specified in subsection 5.1.1.1, the road used for the tests can be approximated as straight with a constant lane width. The road frame is therefore created by arbitrarily setting a point $(x_{0,\mathscr{R}}, y_{0,\mathscr{R}}) = (0, 0)$ on the right mark at the beginning of the demonstration portion, which corresponds to the $\mathscr{W}$-coordinate latitude $\text{Lat}_0 = 48.78513748°$, longitude $\text{Long}_0 = 2.100552545°$, orientation $\theta_0 = 158.40035°$. The x-orientation of the road frame is aligned with the mostright lane mark. The y-coordinate is positive in the direction of the mostleft lane mark.

- Ego frame $\mathscr{E}$: This frame is fixed in the middle of the rear axle tree, with the x-direction following the orientation of the ego vehicle and the y-direction to the left side of the ego vehicle. The angle with $\mathscr{R}$ is defined by α in Figure 5.3.

- Obstacle frame $\mathscr{O}$: It has the same properties as $\mathscr{E}$ by replacing the ego vehicle with an obstacle vehicle.

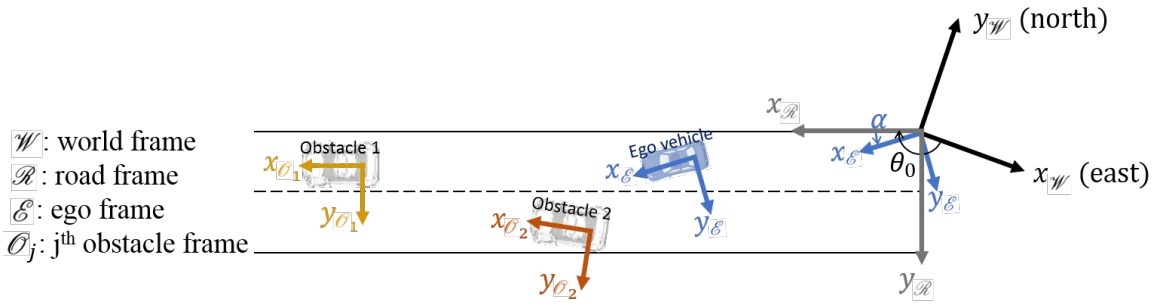


Figure 5.3: Frames representation.

The coordinate transformations for each of the frames are the following:

- $\mathscr{W} \to \mathscr{R}$:

$$\begin{cases} x_{\mathscr{R}} = x_{\mathscr{W}} \cos(-\theta_0) + y_{\mathscr{W}} \sin(-\theta_0) \\ y_{\mathscr{R}} = -x_{\mathscr{W}} \sin(-\theta_0) + y_{\mathscr{W}} \cos(-\theta_0) \end{cases} \tag{5.1}$$

- $\mathscr{R} \to \mathscr{E}$:

$$\begin{cases} x_{\mathscr{E}} = x_{\mathscr{R}} \cos(\alpha) + y_{\mathscr{R}} \sin(\alpha) \\ y_{\mathscr{E}} = -x_{\mathscr{R}} \sin(\alpha) + y_{\mathscr{R}} \cos(\alpha) \end{cases} \tag{5.2}$$

- $\mathscr{O} \to \mathscr{E}$:

$$\begin{cases} x_{rel,O_j\,\mathscr{E}} = x_{O_j,\mathscr{R}} - x_{ego,\mathscr{R}} \\ y_{rel,O_j\,\mathscr{E}} = y_{O_j,\mathscr{R}} - y_{ego,\mathscr{R}} \end{cases} \tag{5.3}$$

5.1.2 Architecture description

This subsection details the architecture and the main components of the implementation of our algorithm. The operating diagram modeled for integrating our motion planner is first exposed in subsection 5.1.2.1, then each block is detailed: the perception block in subsection 5.1.2.2, the motion planner in subsection 5.1.2.3 and the control block in subsection 5.1.2.4. Finally, the data sources are listed in subsection 5.1.2.5.

5.1.2.1 Operating diagram

In this section, we describe the different blocks areas considered in the operating diagram displayed in Figure 5.4. We implemented a localization area with the IMU ego information, a communication area with a UHF receiver, a perception area with the LIDAR data, our motion planner, and a control area separated with a longitudinal velocity controller and a lateral angle controller. The motion planner is integrated through a MatLab/Simulink Coder in the framework of RTMaps (Real Time, Multisensor, Advanced Prototyping Software). There is no specific strategy for synchronization and latency (up to 500 ms for the localization block and 300 ms for the perception block).

Ego data The ego data longitudinal and lateral positions and heading angle are acquired by the IXBLUE ATLANS-C system (`SocketAtlans` and `AtlansDecoder` blocks) in the world frame $\mathscr{W}$. They are vectorized into 1 ego vector (`Ego_Vect`) and transformed through `AtlansToXYV_1` block into the road frame $\mathscr{R}$ as a vector with the longitudinal and lateral positions and longitudinal velocity $\chi_{ego} = (x_{ego}, y_{ego}, v_{ego})_{\mathscr{R}}$, which is then sent to the motion planner block.

Communication obstacle data The longitudinal and lateral positions and heading angle acquired from the IXBLUE ATLANS-C system for the first obstacle O_1 are sent to the ego vehicle using the UHF communication module (`SerialPort_UHF`, `Stream2Text_1`, and `Vedecom_UHF_Decoder_1`). As for the ego data, this information is transformed through `AtlansToXYV_2` block from $\mathscr{W}$ into $\mathscr{R}$ as $\chi_{O_1} = (x_{O_1}, y_{O_1}, v_{O_1})_{\mathscr{R}}$, and sent to the block `Obst_Vectorizer`.

LIDAR detection obstacle data The data for the second obstacle O_2 are obtained from the LIDAR sensors with a 4 layers Ibeo fusion box (`FusionBox` and `IbeoEcuDecoder_1` blocks). The 3D LIDAR data are XYZ point clouds obtained in the ego frame $\mathscr{E}$. Only data from the second bottom layer are selected (`custom_cutter_1`) in order to send 2D data XY to the in-house algorithm `VEDECOM_DeTroLi_Transdev_1`. DetroLi algorithm detects obstacles as points clouds and returns ego relative bounding box around obstacles in $\mathscr{E}$ coordinate systems. The obstacle of interest O_2 is then manually selected in `DetroliFilter_1` and transformed into $\mathscr{W}$ coordinates using the ego coordinates in $\mathscr{W}$. Finally, its coordinates in $\mathscr{W}$ are transformed through `AtlansToXYV_3` block into $\mathscr{R}$ as $\chi_{O_2} = (x_{O_1}, y_{O_1}, v_{O_1})_{\mathscr{R}}$, and sent to the `Obst_Vectorizer` block.

Obstacles The obstacles vectors χ_{O_1} and χ_{O_2} are vectorized (`Obst_Vectorizer`), with a simulated third obstacle (`Obst`) not used in our experimentations, into 1 output obstacle vector, which is sent to the motion planner.

Planning The developed motion planner (`Planner`) with the decision-maker and trajectory generator uses χ_{ego}, χ_{O_1}, χ_{O_2}, and the tuning parameter (`PP_Params`) to calculate the reference trajectory over the prediction horizon H_p in $\mathscr{E}$: $\chi_{ref} = (x(t), y(t), v(t))_{\mathscr{E},(t=1..H_p)}$.

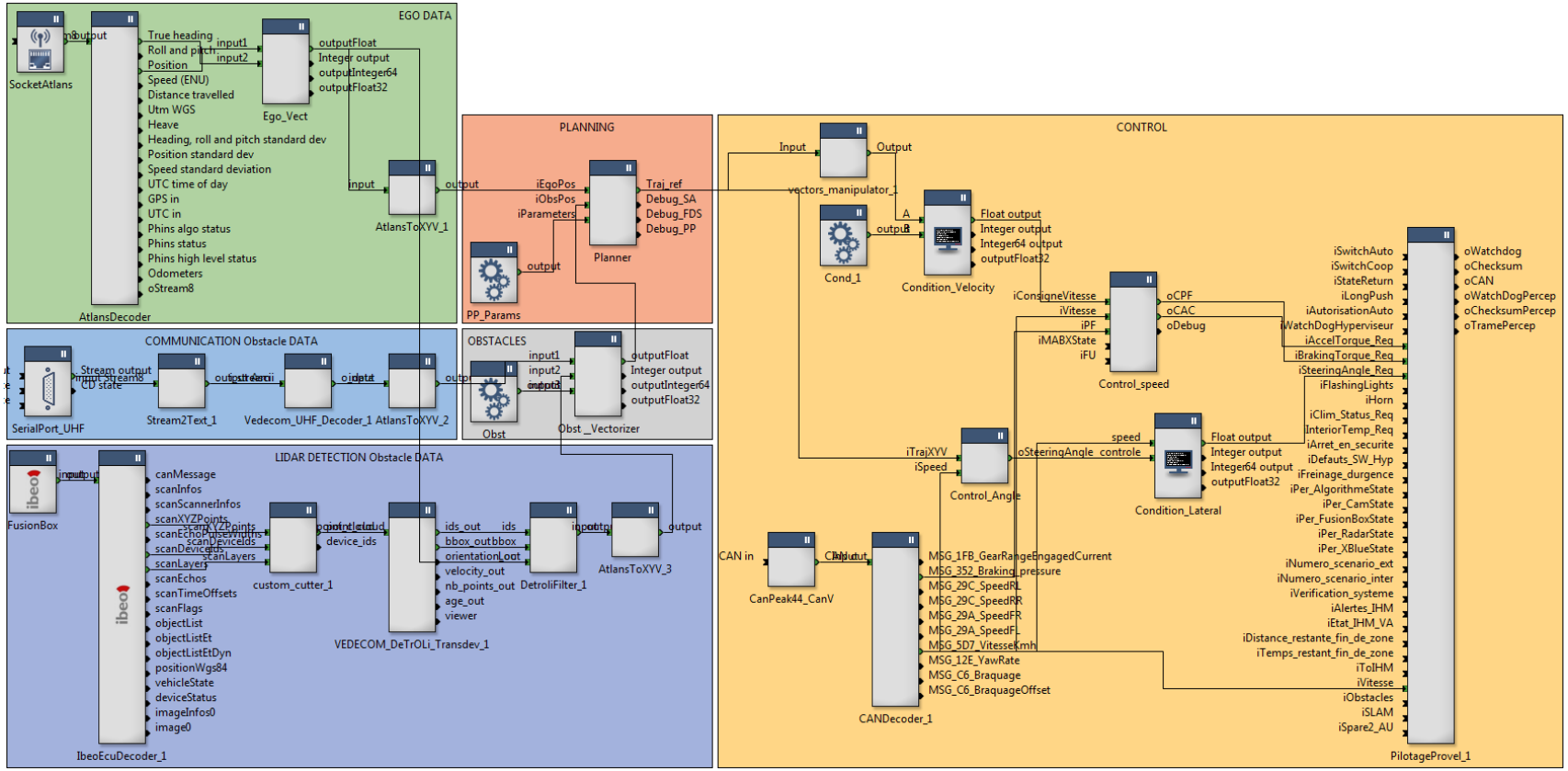


Figure 5.4: RTMaps diagram. The ego, obstacle communication, and obstacle perception data are issued respectively from the green, blue, and purple areas. All the obstacles data are gathered in the gray area to return one obstacle vector. The motion planner is then activated in the orange area and returns the reference trajectory to the controllers in the yellow area.

Control The reference trajectory is sent to the lateral angle controller (`Control_Angle`) and the reference velocity vector is sent to the longitudinal velocity controller (`Control_speed`). The reference velocity is sent to the controller block if (`Condition_Velocity`) its velocity value at 1.5 s ahead (see subsection 5.1.2.4 for more details) is higher than the threshold condition (`Cond_1` equal to 3.6 km/h). Then the `Control_speed` block calculates the acceleration and breaking torques to send to the actuators (`PilotageProvel_1`) according to the control reference velocity value, the current ego velocity and the current breaking pressure, issued from the Controller Area Network (CAN bus: `CANPeak44_CanV` and `CANDecoder_1`), otherwise the value 1 m/s is sent. The reference trajectory and the current ego velocity are sent to the lateral angle controller block to return the steering angle command. If (`Condition_Lateral`) the current ego velocity is higher than 7.2 km/h, this steering angle command is sent to the actuators (`PilotageProvel_1`), otherwise the value 0 rad is sent. The condition values are used at the initialization step to activate the motion planner only at nominal driving conditions.

5.1.2.2 Perception block

The scene representation feeds the motion planner with a representation of the road, obstacles, and ego vehicle. The required information to navigate in the environment concern the lane markings, the road limits, the road speed limits, and the positions/velocities of the obstacles/ego vehicle. The road parameters are fixed offline, as described in subsection 5.1.1.1. The ego vehicle and obstacle O_1 localizations are respectively described in subsections 5.1.1.2 and 5.1.1.3. In this subsection, we only detail the perception algorithm DeTroLi for obstacle O_2.

The position and orientation of O_2 are acquired by the Ibeo fusion box with 4 layers. The 3D LIDAR data are XYZ point clouds obtained in the ego frame $\mathscr{E}$. DeTroLi uses only 1 layer with 2D XY point cloud. It consists in a succession of three steps: (i) point cloud segmentation, (ii) association, and (iii) dynamics estimation.

The first step corresponds to the extraction of obstacles from the raw LIDAR data. It is performed by constructing the Delaunay triangulation of the cloud, trimming edges in the triangulation using a distance metric adapted to the projective properties of the sensor, and extracting the connected components in the resulting graph. In the second step, previously observed obstacles are matched to new observations using a Hungarian algorithm with a geometric distance-based loss function. For the last step, a combination of Iterative Closest Point (ICP) and high-level features fitting is used to estimate the kinematics of the tracked obstacles. The resulting obstacles representation is illustrated in Figure 5.5.

The main limitation of this algorithm is due to the limitation of the data it uses. In particular, when using a LIDAR, the further the observed environment is, the lesser the points density is. Therefore, precise kinematics estimation is particularly complicated at distances above 50 m using readily available LIDAR. This distance limitation can be overtaken with radar sensors for future experimentations.

5.1.2.3 Planning block

The planning block relies on the information of the map, namely lane markings, road limits, and speed limits (defined in the parameters input), the ego vehicle vector $\chi_{ego} = (x_{ego}, y_{ego}, v_{ego})_{\mathscr{R}}$

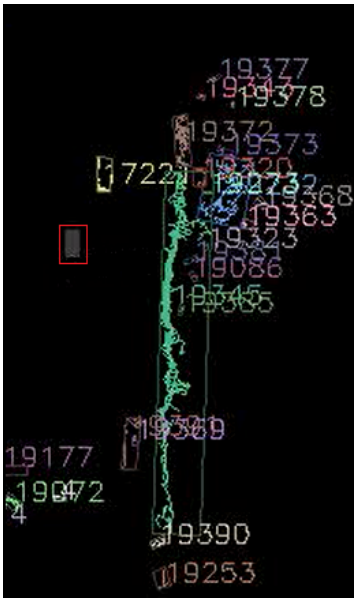


Figure 5.5: DeTroLi viewer. The ego vehicle is the gray rectangle with red edges, the tracked obstacle is number 722.

and obstacles vector $\chi_{O_j} = (x_{O_j}, y_{O_j}, v_{O_j})_{\mathscr{E}}$. It uses these information to return the reference trajectory over the predicted horizon H_p, $\chi_{ref} = (x(t), y(t), v(t))_{\mathscr{E},(t=1..H_p)}$ to the control block. The trajectory is planned over a time horizon $H_p = 10$ s with a time step $\Delta t = 0.05$ s, generating 200 coordinates in $\mathscr{E}$ starting at the current state of the ego vehicle. As explained in subsection 5.1.1.3, we consider maintaining direction and velocity for obstacles prediction.

The pseudo-code of our algorithm is presented in Algorithms 4, 5 and 6. In addition to the assumptions for the nominal use in subsection 4.4.1 (p. 93), we defined the following experimental assumptions for the integration of our algorithm:

Assumption 7 *The motion planner is triggered every* 60 ms*, which corresponds to the maximum calculation time in simulation, or if an obstacle enters the safety space as defined in Figure 5.6.*

Assumption 8 *The motion planner calculates a new trajectory as soon as one of the surrounding obstacles enters the safety space around the ego vehicle, as illustrated in Figure 5.6.*

Assumption 9 *The motion planner is in a default mode if it can not find a valid solution or if there is an obstacle detected inside its collision space, as illustrated in Figure 5.6.*

The default mode is a non-nominal mode. In our experimentations, the default reference trajectory corresponds to stopping in the right road limit with a constant longitudinal deceleration $d_x = -0.7$ m/s^2, i.e. $\chi_{ref,default} = (0, 0, 0)$. This default trajectory is thus easily identifiable by the driver.

Assumption 10 *While the ego vehicle changes lane, the motion planner does not make any decision.*

This assumption has to be removed in the future, so that the motion planner is reactive to its surroundings.

Assumption 11 *If there is no obstacle, the two authorized maneuvers are either to maintain direction and velocity or to move to the right lane according to the French Driver Rules.*

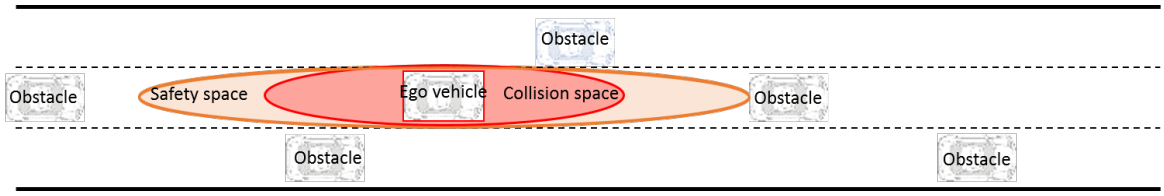


Figure 5.6: Safety and collision spaces around the ego vehicle.

For the experimentation, we use the Mamdani Fuzzy Inference System (FIS) from category 3 – Obstacle safety, in Table 3.1 (p. 55). Its properties are written in Table 5.1, its membership functions μ are plotted in Figure 5.7, its output values are plotted against 2 outputs in Figure 5.22, and its rule-base is depicted in Table 3.4 (p. 78).

Based on the simulation tests in section 4.6, the simulated annealing (SA) algorithm implemented for the experimental results and presented in Algorithm 3 (p. 101) has the following parameters: $N_{\text{iter}} = 200$, $T_0 = 0.6213$, $T_f = 0.01$, and $P_0 = 20\%$ with $Z_0 = 1$.

5.1.2.4 Control block

A control block has been implemented to follow the reference trajectory sent by the motion planner. In order to integrate in the current control architecture of the test vehicle, the longitudinal and lateral control are considered separately. The longitudinal control is based on the longitudinal velocity, and the lateral control is based on the angular error. The two controllers are effective when the ego velocity v_{ego} is over 3.6 km/h for the velocity controller and 7.2 km/h for the angle controller.

Longitudinal controller The longitudinal velocity control is a Proportional Integral (PI) based on the reference velocity tracked 1.5 s ahead. It returns the output torque to the actuators. The control architecture separates acceleration and breaking, see Figure 5.8. Besides, the acceleration control is split into command tracking, i.e. transient response with reference velocity changes, and regulation, i.e. staying at a given reference velocity with disturbance rejection. The value of the parameters have been manually tuned based on experimentations and are gathered in Table 5.2.

Lateral controller The lateral control is a proportional gain (P) on the lateral gap e_{gap} and the cap angle α_{cap} to return the steering angle command α_c. The control low is expressed in Equation 5.4.

$$\alpha_c = K_e \cdot e_{gap} + K_\alpha \cdot \alpha_{cap} \tag{5.4}$$

where K_e is the lateral gap proportional gain and K_α is the cap angle proportional gain.

Figure 5.9 shows how e_{gap} and α_{cap} are calculated. A distance target d_{target} within the range $[d_{target,min}, d_{target,max}]$ is defined with a parameter k_v proportionally to the ego velocity v_{ego}: $d_{target} = k_v \cdot v_{ego}$. The closest reference points from χ_{ref} are identified as

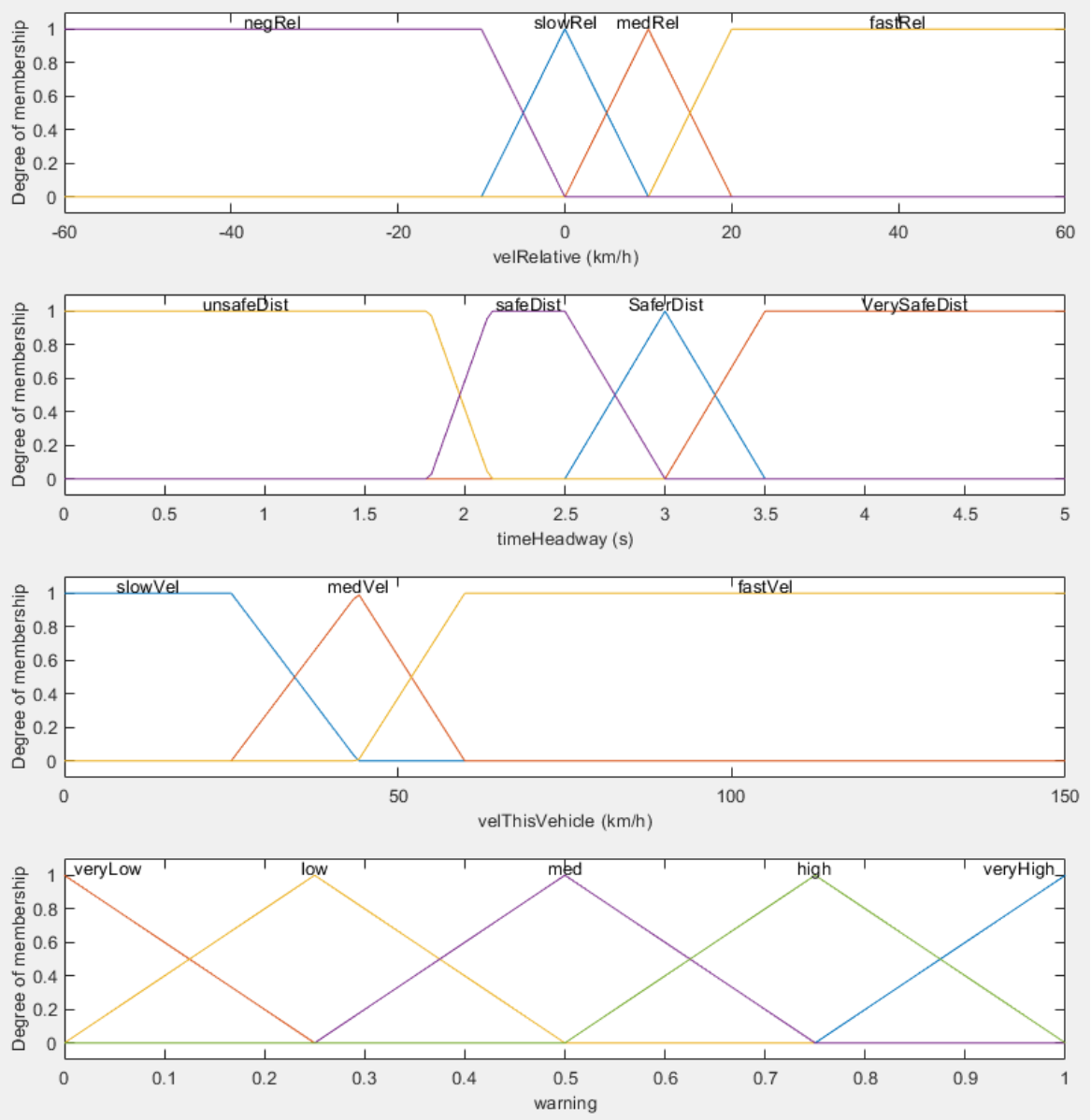


Figure 5.7: FIS inputs and output membership functions μ.

p_{ref} and p_{ref-1}. The target point p_{target} is interpolated within $[p_{ref-1}\ p_{ref}]$. The lateral distance between p_{target} and y-axis in $\mathscr{E}$ is equal to e_{gap} and the angle formed by the segment $[p_{ref-1}\ p_{ref}]$ with the x-axis in $\mathscr{E}$ is equal to α_{cap} expressed in Equation 5.5:

$$\alpha_{cap} = atan\left(\frac{y_{p_{ref},\mathscr{E}} - y_{p_{ref-1},\mathscr{E}}}{x_{p_{ref},\mathscr{E}} - x_{p_{ref-1},\mathscr{E}}}\right). \tag{5.5}$$

The parameters are defined experimentally: $d_{target,min} = 5$ m, $d_{target,max} = 60$ m, $k_v = 1.5$ s, $K_e = 0.2$ m^{-1}, and $K_{cap} = 1$.

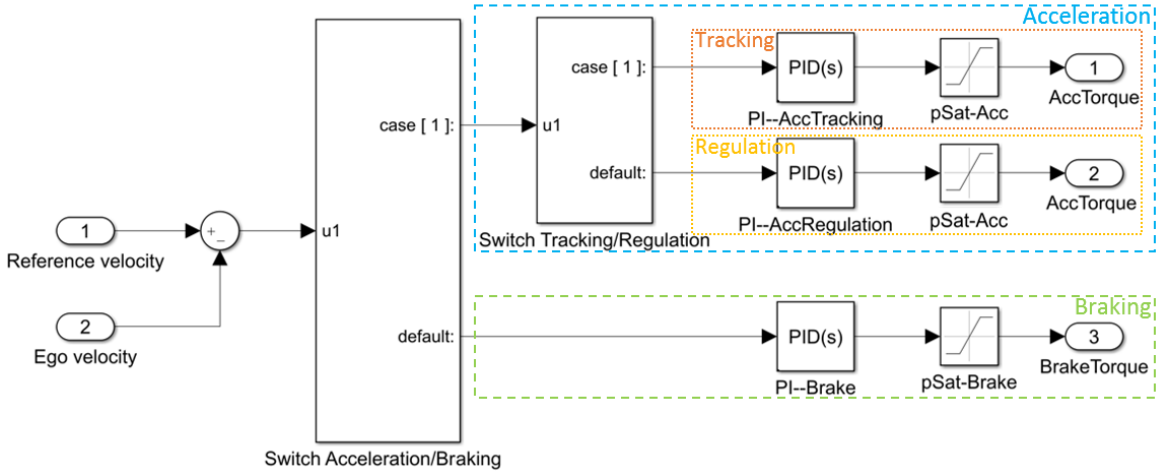


Figure 5.8: PI architecture for the longitudinal controller.

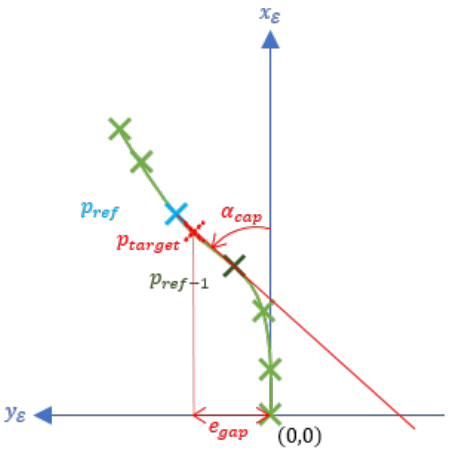


Figure 5.9: Variables illustration for the lateral controller.

Algorithm 4: Main pseudo-code

Data: Scene representation
Result: Reference trajectory
Initialization and update;
/* 1. Calculate the trajectory in $\mathscr{R}$ */
if *not currently lane changing* // Lane changing condition
then
  if *no obstacle* // Obstacle condition
  then
    Return maintain wish velocity and direction or apply driving rules (e.g. go right);
  else
    if *safe distance* // Obstacle condition
    then
      Return maintain wish velocity and direction or apply driving rules (e.g. go right);
    else
      Define ego profiles $p_{i=1..N}$;
      Define obstacles profiles $p_{O_{j=1..J}}$;
      Define intervals $I_{k=1..K} \equiv (O^{inf}, O^{sup}, dir)$;
      Test collision ego profiles in velocity and distance;
      Characterize intervals $(v_{inf}, v_{sup}, d_{inf}, d_{sup})$;
      Calculate optimal trajectory for staying in ego lane; // Straight direction pseudo-code
      if *direction exists* // Test adjacent lane existences
      then
        Calculate optimal trajectory for lane changing; // Left/Right direction pseudo-code
      else
        Discard direction;
      Calculate the best evaluation indicator among the remaining directions;
      Return best trajectory;
else
  Update the current lane change trajectory;
/* 2. Check trajectory safety */
/* 2.1 Stay within road limits condition */
/* 2.2 Exist trajectory condition */
if *no 2.1 || no 2.2* **then**
  Return default trajectory;
else
  Return the best trajectory;
/* 3. Transform the road frame trajectory in the ego frame trajectory */
Transform the reference trajectory $\mathscr{R} \rightarrow \mathscr{E}$;

Algorithm 5: Straight direction pseudo-code

```
Data: Intervals bounds
Result: Optimized straight trajectory for staying maneuver
Initialization;
if no collision-free ego profile then
    Return no solution to go straight;
else
    Calculate the initial solution;
    Run the optimization process SA;                    // SA pseudo-code
Return the optimized straight trajectory or no solution;
```

Algorithm 6: Left/Right direction pseudo-code

```
Data: Intervals bounds
Result: Optimized left/right trajectory for left/right lane changing maneuver
Initialization;
/* 1.  Select the left/right intervals                                  */
for each interval I_k do
    /* 1.2 Get the collision-free ego profile                           */
    if no collision-free ego profile then
        Return no solution to go left/right;
    else
        if no ego profile for the maneuver    // Test maneuver gathering with I_1
         then
            Discard maneuver;
        else
            Calculate the SA bounds for the variables (v, c, λ) for each remaining ego
             profile;
            Calculate the initial solution;
            Run the optimization process SA;                // SA pseudo-code
    Return the optimized left/right trajectory or no solution;
Select the best maneuver among the best maneuvers for each intervals;
Return the best optimized left/right change trajectory or no solution;
```

Table 5.1: FIS used in experiments.

Type:	Mamdani
AndMethod:	*min*
OrMethod:	*max*
ImplicationMethod:	*min*
AggregationMethod:	*max*
Inputs:	3
Outputs:	1
Rules:	48

Table 5.2: Longitudinal PI controller parameters.

Parameter	**Value**	**unit**
`PI-AccRegulation-Kp`	-128.0	N.s
`PI-AccRegulation-Ki`	-5.0	N^{-1}
`PI-AccTracking-Kp`	-50.0	N.s
`PI-AccTracking-Ki`	-5.0	N^{-1}
`pSatdown-Acc`	0.0	N.m
`pSatup-Acc`	500.0	N.m
`PI-Brake-Kp`	-250.0	N.s
`PI-Brake-Ki`	-150.0	N^{-1}
`pSatdown-Brake`	-10000.0	N.m
`pSatup-Brake`	0 or 5000.0	N.m

5.1.2.5 Data sources

The input and output data previously described for the motion planner are summarized in Table 5.3.

Table 5.3: Data sources (i: input, o: output).

Variable	Definition	Sources	Post-processing
i: $\chi_{ego} = \begin{cases} x_{ego,\mathscr{R}} \\ y_{ego,\mathscr{R}} \\ v_{ego,\mathscr{R}} \end{cases}$	longitudinal position of ego vehicle in $\mathscr{R}$	ATLANS (longitude, latitude, orientation)$_{\mathscr{W}}$	Convert from $\mathscr{W}$ to $\mathscr{R}$
	lateral position of ego vehicle in $\mathscr{R}$	ATLANS (longitude, latitude, orientation)$_{\mathscr{W}}$	Convert from $\mathscr{W}$ to $\mathscr{R}$
	longitudinal velocity of ego vehicle in $\mathscr{R}$	/	Differentiation of $x_{ego,\mathscr{R}}$ and $y_{ego,\mathscr{R}}$
i: $\chi_{O_1} = \begin{cases} x_{O_1,\mathscr{R}} \\ y_{O_1,\mathscr{R}} \\ v_{O_1,\mathscr{R}} \end{cases}$	longitudinal position of obstacle 1 in $\mathscr{R}$	ATLANS (longitude, latitude, orientation)$_{\mathscr{W}}$	Convert from $\mathscr{W}$ to $\mathscr{R}$
	lateral position of obstacle 1 in $\mathscr{R}$	ATLANS (longitude, latitude, orientation)$_{\mathscr{W}}$	Convert from $\mathscr{W}$ to $\mathscr{R}$
	longitudinal velocity of obstacle 1 in $\mathscr{R}$	/	Differentiation of $x_{O_1,\mathscr{R}}$ and $y_{O_1,\mathscr{R}}$
i: $\chi_{O_2} = \begin{cases} x_{O_2,\mathscr{R}} \\ y_{O_2,\mathscr{R}} \\ v_{O_2,\mathscr{R}} \end{cases}$	longitudinal position of obstacle 2 in $\mathscr{R}$	LIDAR Ibeo Fusion Box in $\mathscr{E}$	Detroli + Filter the obstacle of interest + Convert from $\mathscr{E}$ to $\mathscr{W}$ to $\mathscr{R}$
	lateral position of obstacle 2 in $\mathscr{R}$	LIDAR Ibeo Fusion Box in $\mathscr{E}$	Detroli + Filter the obstacle of interest + Convert from $\mathscr{E}$ to $\mathscr{W}$ to $\mathscr{R}$
	longitudinal velocity of obstacle 2 in $\mathscr{R}$	/	Differentiation of $x_{O_2,\mathscr{R}}$ and $y_{O_2,\mathscr{R}}$
i: road	lane coordinate, lane width, number of lane, speed limits	apriori fixed	/
o: $\chi_{ref} = \begin{cases} x^{ref}_{ego,\mathscr{E}} \\ y^{ref}_{ego,\mathscr{E}} \\ v^{ref}_{ego,\mathscr{E}} \end{cases}$	longitudinal position of ego vehicle in $\mathscr{E}$ along H_p	/	Convert from $\mathscr{R}$ to $\mathscr{E}$
	lateral position of ego vehicle in the ego frame $\mathscr{E}$ along H_p	/	Convert from $\mathscr{R}$ to $\mathscr{E}$
	longitudinal velocity of ego vehicle in the ego frame $\mathscr{E}$ along H_p	/	Convert from $\mathscr{R}$ to $\mathscr{E}$

5.2 Experimental scenarios and results

To validate our approach, we consider the following highway driving situations among the ones listed in subsection 2.1.3 (p. 18): lane keeping, car following, lane changing, lateral-most lane changing, passing, and overtaking. We distinguish 6 scenarios to propose realistic experimental conditions:

- Scenario 1: Road following, Lane keeping and Lane changing in subsection 5.2.1;
- Scenario 2: Vehicle following in subsection 5.2.2;
- Scenario 3: Overtaking one front obstacle in subsection 5.2.3;
- Scenario 4: Overtaking two front obstacles in subsection 5.2.4;
- Scenario 5: Overtaking the front obstacle with merging (1) in front of, and (2) behind, the left obstacle in subsection 5.2.5.1, respectively 5.2.5.2.

For each scenario, we first present its goals and constraints, and then analyze the experimental data in order to evaluate and validate our algorithm. We run each scenario multiple times to verify its repeatability. As each scenario was actually repeatable, only one run per scenario is analyzed in the following. The algorithm performances are also given in subsection 5.2.6.

The experimental results are presented in two figures: the ego trajectories and limits, and the relative velocities and positions with the motion planner behavior. The first figure is split into four plots:

(a) shows the reference χ_{ref}, ego χ_{ego}, and obstacles χ_{O_1}, χ_{O_2} trajectories in space and time in the road frame $\mathscr{R}$;

(b) gives the evolution of the velocities with time: the driver desired velocity v_{wish}, the road speed minimum v_{min} and maximum v_{max} limits, the reference velocity v_{ref} from χ_{ref}, the ego velocity v_{ego}, and the obstacles velocities v_{O_1}, v_{O_2};

(c) displays the ego accelerations evolution in time with the longitudinal a_x and lateral a_y ego accelerations. The attainable longitudinal accelerations given in [Reif & Dietsche 2010] lie within the range of $0.4 - 1.4$ m/s^2. We thus considered a reference longitudinal acceleration, respectively deceleration, value of $a_{x,ref} = 0.9$ m/s^2, respectively $d_{x,ref} = -0.7$ m/s^2. In [Reif & Dietsche 2010], the range for lateral acceleration lies within $[-2, 2]$ m/s^2. These limits are represented with $d_{y,lim}$ and $a_{y,lim}$;

(d) provides the time evolution of the steering angle and the yaw rate.

The second figure is split into four plots:

(a) shows the relative velocities between the ego vehicle and O_1, v_{rel,O_1}, and O_2, v_{rel,O_2}.

(b) gives the longitudinal relative distances between the ego vehicle and O_1, x_{rel,O_1}, and O_2, x_{rel,O_2}. The ego vehicle presents safe distances with obstacle O_j, if their relative longitudinal distance x_{rel,O_j} in front, respectively rear, is bigger than the front safe

distance $d_{f,safe}$, respectively the rear safe distance $d_{r,safe}$. Similarly, the ego vehicle is in collision with obstacle O_j, if their relative longitudinal distance x_{rel,O_j} in front, respectively rear, is smaller than the front collision distance $d_{f,collision}$, respectively the rear collision distance $d_{r,collision}$. Overcoming the safe distances leads the triggering of the motion planner. Overcoming the collision distances leads the default mode.

(c) displays the lateral relative distances between ego vehicle and O_1, y_{rel,O_1}, and O_2, y_{rel,O_2}.

(d) plots the motion planner flags to verify a situation of collision, the decisions of lane changing and navigating back to the right lane, if there is no solution, and if the ego vehicle is in default mode navigating off the road limits.

5.2.1 Scenario 1: Road following, Lane keeping and Lane changing

The first scenario consists in verifying the performances of the control block in order to make sure that the reference trajectory is feasible and the control loop is good enough before testing more complex scenarios. We first test a situation of road following and lane keeping, and then a lane change without obstacle.

The first situation is illustrated in Figure 5.10. It corresponds to a road following and lane keeping behaviors without obstacle. The ego vehicle must stay within the ego lane markings and follow the reference velocity v_{ref}. v_{ref} is calculated by the motion planner to follow the wish velocity v_{wish} within the speed limits v_{min} and v_{max}. In this situation v_{wish} can be a preference velocity at which the driver desires to navigate. v_{wish} is modified as a motion planner parameter and will increases and decreases along the scenario at t_1 and t_2.

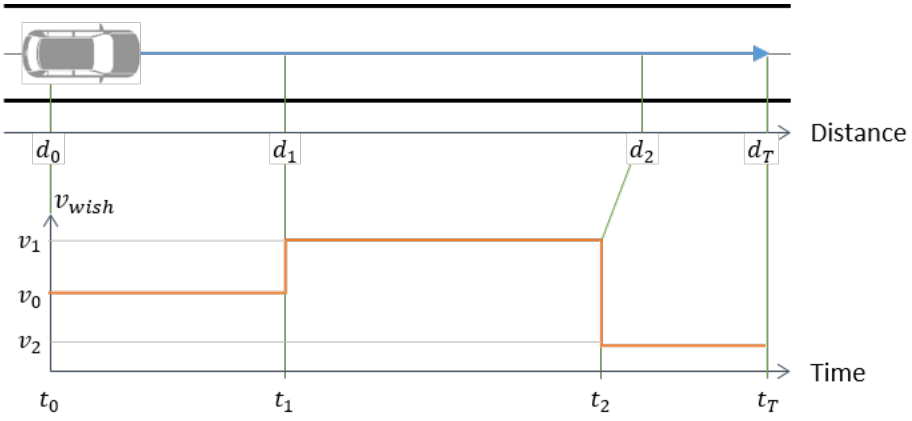


Figure 5.10: Illustration of scenario 1.

Figure 5.11a shows that the reference trajectory χ_{ref} is well followed by the ego vehicle χ_{ego} both in space and time. The longitudinal distance delay is less than 2 m (data points 1 and 2), the lateral error less than 20 cm (data points 1 and 2), and the time delay less than 200 ms. Moreover, one notices an offset to the right side of the road on the lateral position. This is due to the deformation of the road with curved borders. This is further confirmed in Figure 5.11d, where the mean steering angle is shifted to a positive value. In fact, the angle's control compensates the rightward road camber by pushing the vehicle to the left side.

In Figure 5.11b, the initial v_{wish,t_0} is fixed at 10 m/s, then decreases to $v_{min} = 9$ m/s at $t_2 = 12.84$ s, and increases at $v_{max} = 12$ m/s at $t_3 = 26.1$ s. Its value exceeds the speed limits at $t_4 = 57.06$ s with $v_{wish,t_4} = 15$ m/s. Conversely, it is below the speed limits at $t_5 = 69.36$ s with $v_{wish,t_5} = 6$ m/s. The reference velocity v_{ref} follows the changes of v_{wish}, while maintaining the reference velocity within the speed limits at t_4 and t_5. Besides, v_{ref} is well tracked by the controller, as $v_{ego} \approx v_{ref}$ along the entire scenario. There are response delays around 100 ms in acceleration (data points 3) and 350 ms in braking (data points 4), as well as an overshoot up to 1.7 m/s in acceleration (data point 5) for the velocity tracking. This is due to the fact that the tracked velocity is taken as the value of v_{ref} 1.5 s ahead in order to prevent oscillations, which causes a time delay in the controller response, and to the algorithm latency and the dynamic response of the vehicle. Nonetheless, these discrepancies (response delay and overshoot) remain acceptable. Furthermore, to avoid an important gap to the velocity controller, v_{ref} follows the overshoot of v_{ego} to decrease it to v_{wish} with a feasible braking action.

As noticed in Figure 5.11c, the longitudinal acceleration is bigger than the one fixed by the reference trajectory $a_{x,ref}$, but remains under the maximum comfort value set to 3 m/s^2 by [Reif & Dietsche 2010]. However, no braking action is clearly observed; the vehicle is coasting. The difference in acceleration is mainly attributed to the fact that the control is performed on the velocity and not on the acceleration, and to the latency of the dynamics of the ego vehicle and the implemented architecture. The braking control has a very limited action in our scenario because the test vehicle had a braking control based on large step commands. As our motion planner tried to follow the dynamic of the vehicle, its braking reference was too smooth to be considered by the controller. This control law also explains why overshoots are perceived during acceleration but not during the braking phases.

Finally, the road camber compensation is observed in the rightward-shifted mean steering angle values in Figure 5.11d.

The second situation analyses a simple lane change to the right lane without obstacle. Figure 5.12a shows the lane change trajectory in time for a reference velocity $v_{ref} = 9.92$ m/s. The time delay along the trajectory is very short, less than 200 ms at the road centerline, which corresponds in a distance delay less than 1.6 m (data points 1) and a slight overshoot of 0.25 m to the right at the end of the lane change (data point 2). Figure 5.12b confirms the proper tracking of v_{ref}. Figures 5.12c and d show the feasible lateral dynamics of the ego vehicle, where a_y is within $[d_{y,lim}, a_{y,lim}]$ and both the steering angle and yaw rate are reasonable.

To summarize, the performances of both the motion planner and the controllers are evaluated good enough for the validation of our algorithms. Besides, the controller performances stated in this first scenario are expected for all scenarios. Since the control design is not in the scope of the present work, we do not pretend to provide a better controller for the next scenarios. Please remind the following performances of the controllers:

- the response is stable for both the velocity and angle controllers;
- the steady-state error is very low: the velocity controller has no significant error (< 0.3 m/s) and less than 20 cm for the lateral controller (which is due to the road camber);

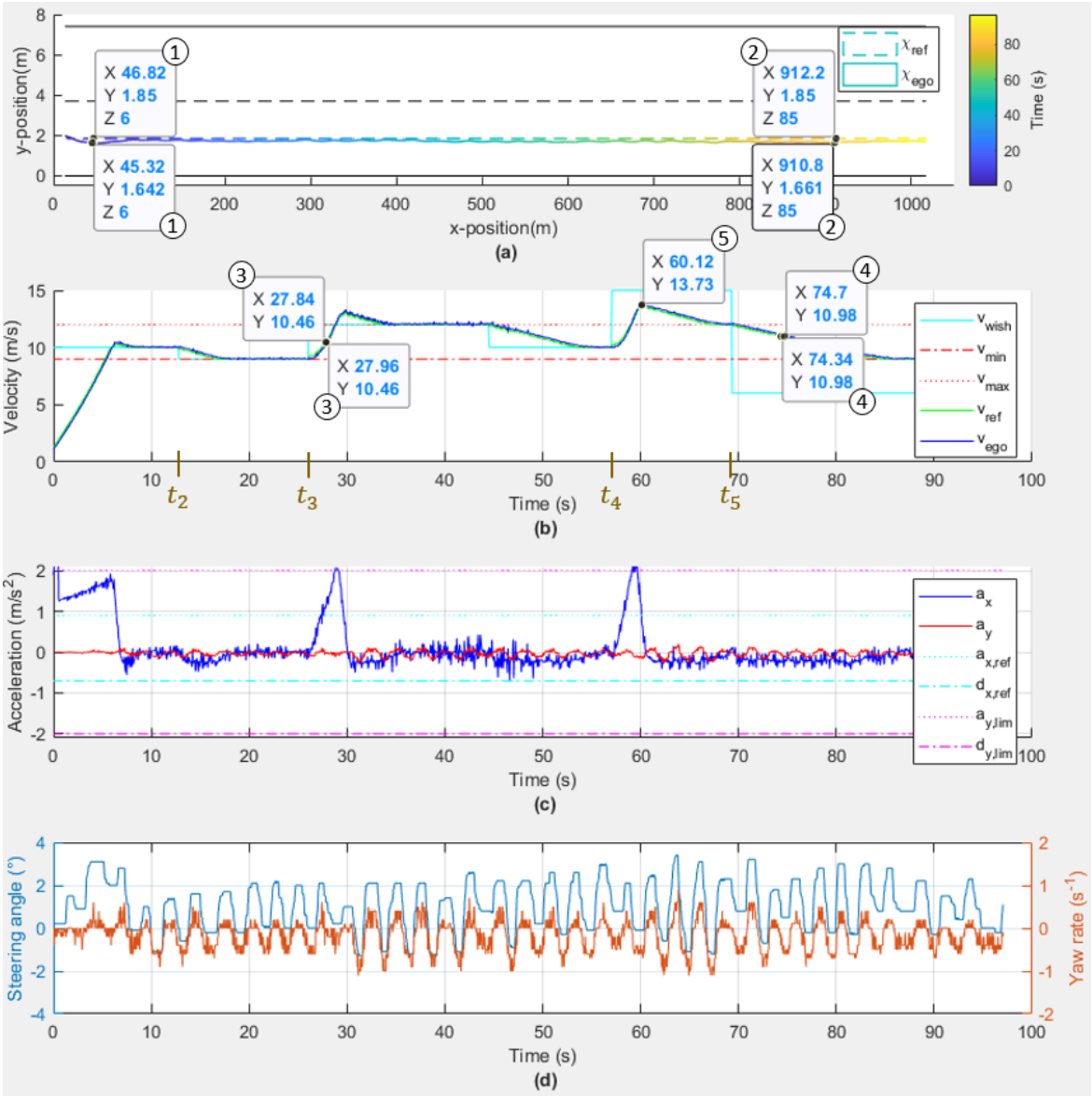


Figure 5.11: Trajectory, velocity and dynamics analysis. In (a), the reference trajectory χ_{ref} is in dashed line, the ego trajectory χ_{ego} in solid line and the obstacles trajectories χ_{O_1}, χ_{O_2} in dashed-dotted lines, if they exist; in (b) the driver wish velocity v_{wish} is in cyan, the road minimum v_{min} and maximum v_{max} speed limits in dotted and dashed red, the reference velocity v_{ref} in solid green, the ego velocity v_{ego} in solid blue, and the obstacles velocities v_{O_1} and v_{O_2} in dashed orange and purple, if they exist; in (c) the ego longitudinal and lateral accelerations a_x and a_y are in solid blue and red, the reference longitudinal acceleration and deceleration $a_{x,ref}$ and $d_{x,ref}$ in dotted and dashed-dotted cyan, and the lateral acceleration and deceleration limits $a_{y,lim}$ and $d_{y,lim}$ in dotted and dashed-dotted magenta; in (d) the steering angle is in blue and the yaw rate in orange. [Scenario 1]

- the overshoot on the longitudinal velocity is between $5-30\%$, and $0-10\%$ for the lateral position, which represents up to 1.5 m/s in velocity and 25 cm in lateral position;
- a time-delay up to 350 ms is observed on the velocity tracking and up to 200 ms on the lateral position tracking. This is due to the control law. Indeed, the tracking was done

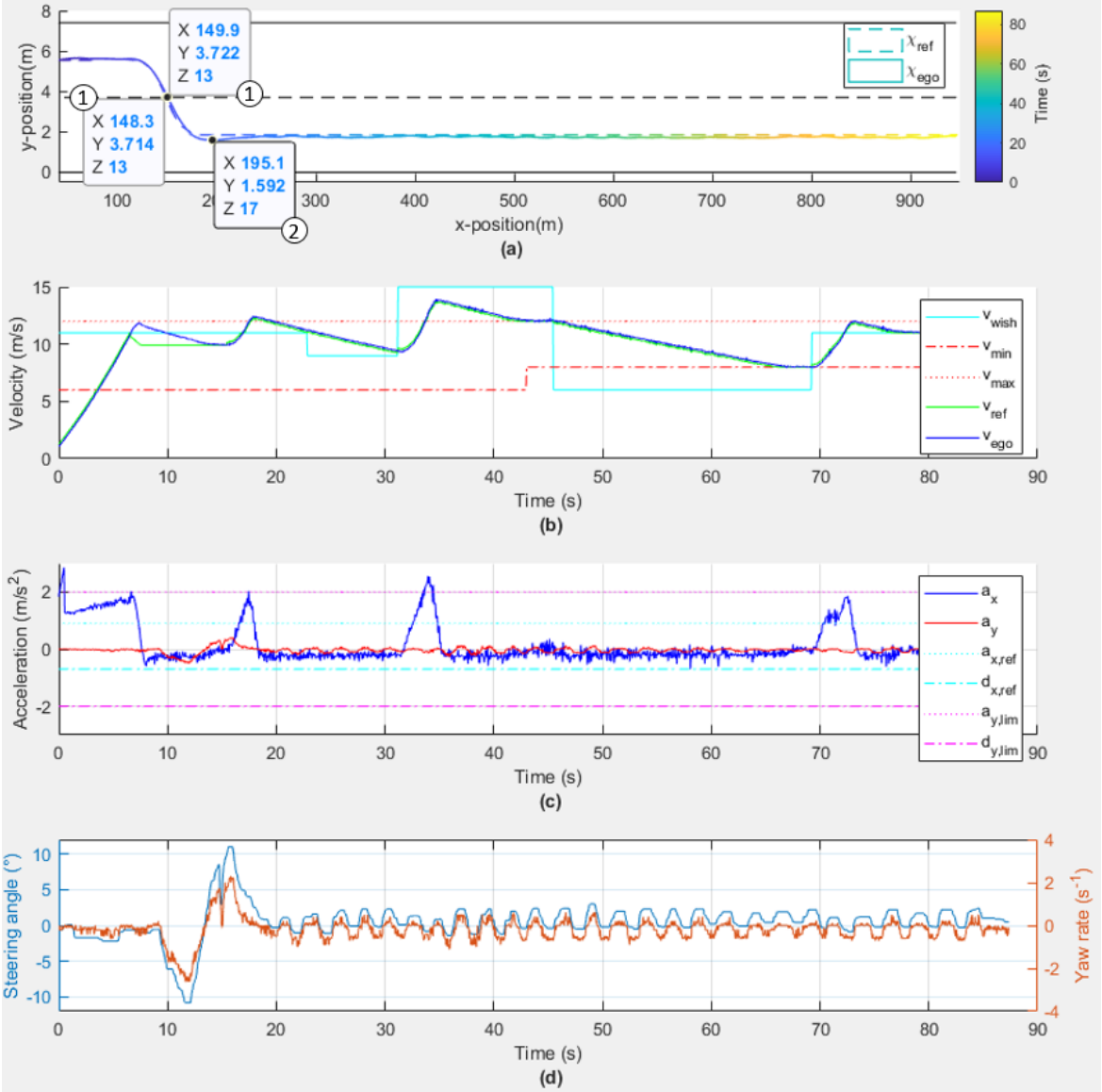


Figure 5.12: Trajectory, velocity and dynamics analysis for a lane changing. [Scenario 1]

1.5 s ahead for the velocity control and the angle control in order to avoid oscillations and over-reaction if the tracking was done on the next time-step value.

5.2.2 Scenario 2: Vehicle following

In this vehicle following scenario illustrated in Figure 5.13, we consider a single lane with a moving front obstacle vehicle O_1. While maintaining the desired position in the centerline of the lane, the ego vehicle adjusts its longitudinal dynamics with O_1 to maintain safety distances and passenger comfort.

The recommendations of the Driver Rules are to maintain a 2 s gap as a human driver with a front obstacle vehicle. During the tests, to ensure the safety of all participants, we set a fixed front, respectively rear, collision distance $d_{f,collision} = 50$ m, $d_{r,collision} = 30$ m,

regardless the velocity of our vehicle v_{ego} and the relative speed with obstacles v_{rel,O_j}. As the maximum allowed velocity in autonomous mode is 50 km/h = 13.9 m/s, this collision distance is overestimated with respect to the 2 s rule. In this scenario, we fixed a safety distance twice bigger than the collision distances, i.e. $d_{f,safe} = 2 * d_{f,collision}$ and $d_{r,safe} = 2 * d_{r,collision}$.

The front moving obstacle O_1 has its initial velocity $v_{0_1,t_0} < v_{ego,t_0}$. Along the scenario, we modified v_{O_1} in order to check that the ego vehicle adapts its velocity to the front vehicle. At the beginning, the obstacle is far enough, out of the safety space of the ego vehicle. The ego vehicle is in the situation of road following. At t_1, the obstacle enters the safety zone of the ego vehicle. The ego vehicle has to slow down and adapts its velocity to the one of the obstacle vehicle v_{O_1,t_1}. At t_2, the obstacle velocity is reduced to $v_{O_1,t_2} < v_{O_1,t_1}$. At t_4, the obstacle accelerates above the maximum speed limit, $v_{O_1,t_4} > v_{max}$. The ego vehicle can then accelerate and reach its wish velocity v_{wish}.

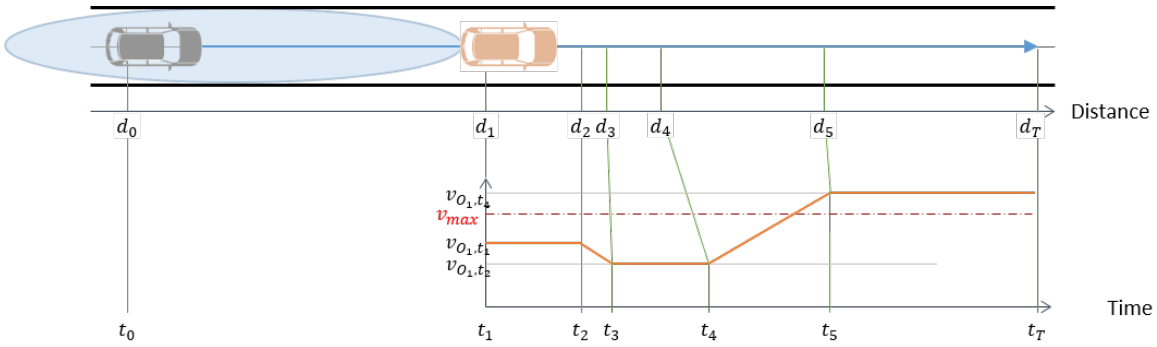


Figure 5.13: Illustration of scenario 2.

This scenario is a tedious task for the motion planner for two reasons: (i) in this situation, the motion planner is triggered at each iteration as the obstacle permanently enters the safety space, (ii) the range of action is very limited to respect the three decision criteria (higher speed, low relative speed, high time headway). We present a nominal behavior of the motion planner and a more critical one, for which we show the limits of our algorithm.

Nominal case: At the beginning of the scenario, the longitudinal relative distance x_{rel,O_1} between the ego vehicle and O_1 is bigger than the front safe distance $d_{f,safe}$ (data points 1 in Figures 5.14a and 5.15b). O_1 enters the safety space at $t_1 = 22.44$ s (data point 2 in Figure 5.15b). Consequently, the flag collision is activated (data point 2 in Figure 5.15d) and the motion planner sends a decelerating v_{ref}, so that the ego vehicle brakes (data point 2 in Figure 5.14b). The ego velocity v_{ego} then decreases until it reaches the velocity which allows to maintain the relative distance x_{rel,O_1} above the collision distance $d_{f,collision}$ (data area 3 in Figures 5.15a and b). The v_{ref} profile is the best compromise of the 3 decision criteria: time headway T_h, relative velocity v_{rel,O_1}, and ego velocity v_{ego}, see subsection 5.1.2.3. Logically, v_{ref} follows v_{O_1} as depicted in data area 3 in Figure 5.14b.

At $t_2 = 50$ s, O_1 increases its velocity up to 10 m/s. Its dynamics can be followed by the ego vehicle, thus the motion planner adapts v_{ref} to v_{O_1} (data area 4 in Figure 5.14b). When v_{O_1} decreases at $t_3 = 64$ s, the motion planner sends a decreasing v_{ref} too (data area 5 in Figure 5.14b). As the dynamics of O_1 is steeper than that of the ego vehicle, one notices that x_{rel,O_1} decreases too (data area 5 in Figure 5.15b). When O_1 increases its velocity at

$t_4 = 83$ s over the maximum speed limit $v_{max} = 12$ m/s, the ego vehicle can also accelerate to its initial wish velocity $v_{wish} = 11$ m/s. If the obstacle velocity is over the maximum speed limit v_{max} (at $t_5 = 85$ s), v_{ref} plateaus at $v_{wish} = 11$ m/s, as seen in Figure 5.14b. O_1 leaves the safety space at $t_6 = 89.58$ s (data points 6 in Figures 5.15b and d).

Moreover, the reference trajectory is well followed in Figure 5.14a. One notices a 10 cm-shift to the right due to the road camber. Slight overshoots ($\approx$ 1m/s) on the velocity control are shown in Figure 5.14b, as explained in scenario 1.

In this scenario, the motion planner has enough time and distance to adapt well to the obstacle dynamics. In Figure 5.15d, one notices that the `no solution` flag is always inactivated, even if the `collision` flag is activated.

Finally, the longitudinal and lateral accelerations and angle constraints for both safety and comfort are respected as shown in Figures 5.14c and d.

Critical case: In the critical case, the minimum speed limit v_{min} is set to 8 m/s, which is the velocity of O_1 at the beginning of the scenario. As a result, there are very small velocity (< 1 m/s) and distance (< 50 m) ranges for the ego vehicle to maintain $x_{rel,O_1} > d_{f,collision}$ (Figure 5.16b). Besides, the motion planner only selects a velocity profile that does not present any collision state along the prediction horizon H_p. Consequently, if the relative velocity and distance do not allow the ego vehicle to reach a collision-free velocity profile before O_1 enters the collision space $x_{rel,O_1} < d_{f,collision}$, none of the allowed velocities are safe. This inconvenience is emphasized with the poor braking capacity of the ego vehicle in our experiments. In this critical case, we thus observe that O_1 enters $d_{f,safe}$ at $t_1 = 20.46$ s and $d_{f,collision}$ after $t_2 = 36.18$ s (data points 1 and 2 in Figures 5.16b and d). The motion planner has difficulty maintaining this collision distance, as a safe velocity profile does not always exist. This is represented by the `no solution` flag in area 2 in Figure 5.16d. In practice, this situation leads to the default mode, with an emergency trajectory tending towards $y_{ref,\mathscr{R}} = 0$ m with a reference velocity profile decreasing to 0 m/s. However, as soon as the collision distance is recovered, the motion planner replans a collision-free velocity profile. O_1 leaves the safety space at $t_3 = 74.58$ s (data points 3 in Figures 5.16b and d).

To conclude, the condition of collision-free ego profile along the prediction horizon may be too strong for real life application, as observed in the critical case. Furthermore, the collision distance should be taken proportional to the ego and relative velocities in future work.

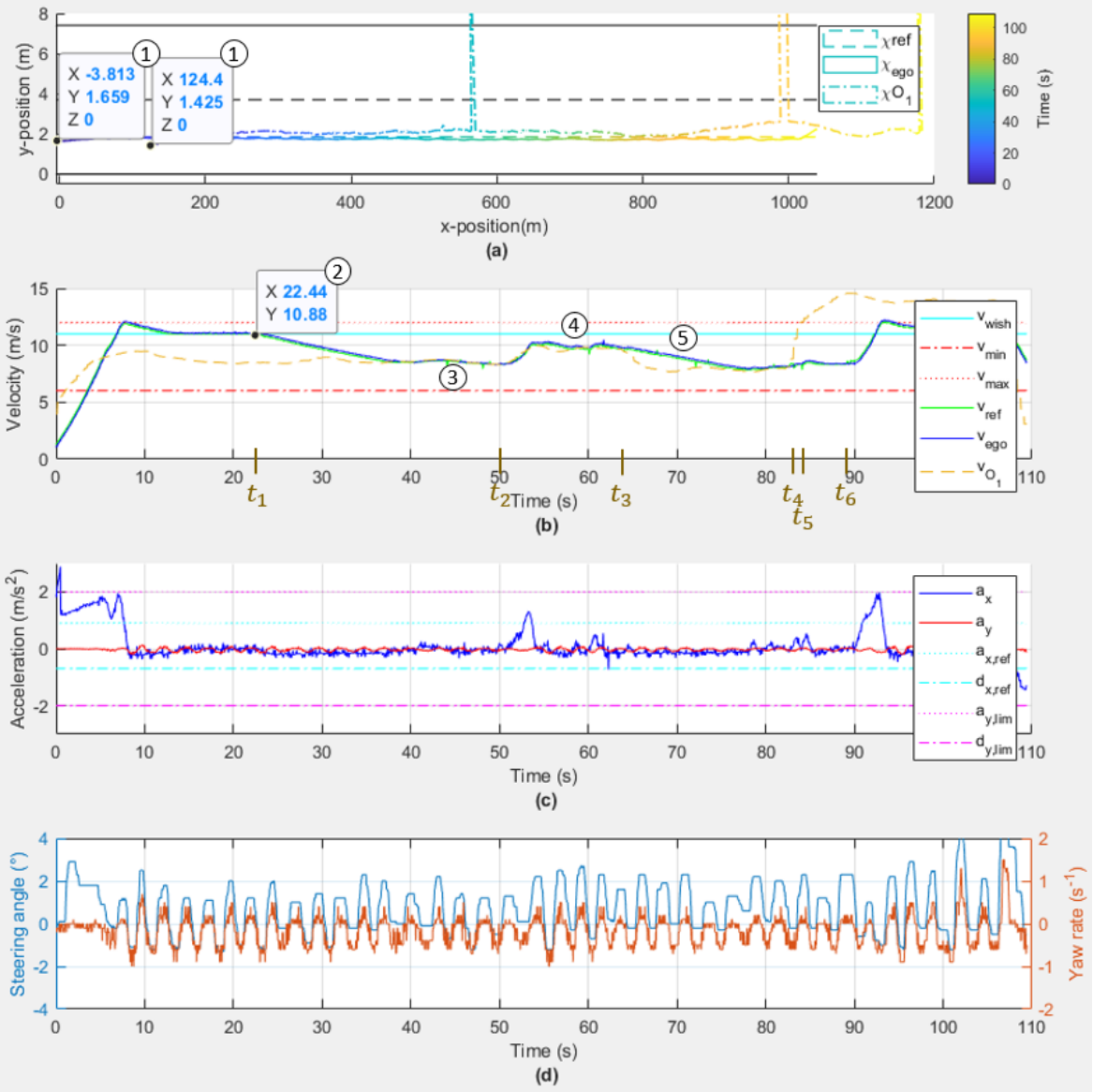


Figure 5.14: Trajectory, velocity and dynamics analysis. The position outliers of O_1 are due to errors in the communication blocks. [Scenario 2]

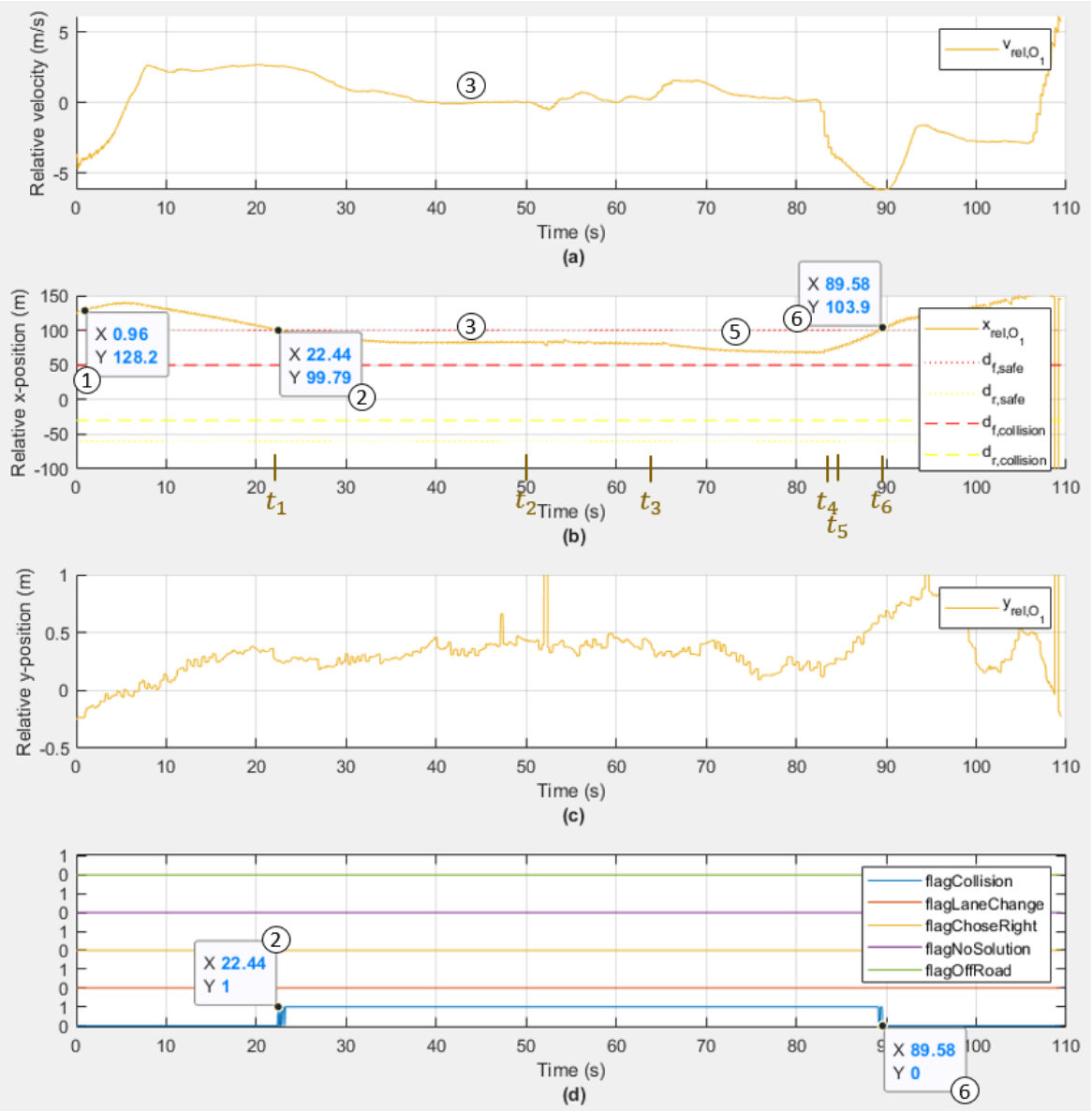


Figure 5.15: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis. The position outliers of O_1 are due to errors in the communication blocks. In (a), the relative velocities v_{rel,O_1} and v_{rel,O_2} are in orange and purple, if they exist; in (b), the longitudinal relative distances x_{rel,O_1} and x_{rel,O_2} are in orange and purple, if they exist, the safe front $d_{f,safe}$ and rear $d_{r,safe}$ distances in dotted red and yellow, the front $d_{f,collision}$ and rear $d_{r,collision}$ collision distances in dashed red and yellow; in (c), the lateral relative distances y_{rel,O_1} and y_{rel,O_2} are in orange and purple, if they exist; in (d) the motion planner flags denote collision in blue, lane changing in red, right lane decision in yellow, no solution in purple and navigating off the road limits in green. [Scenario 2]

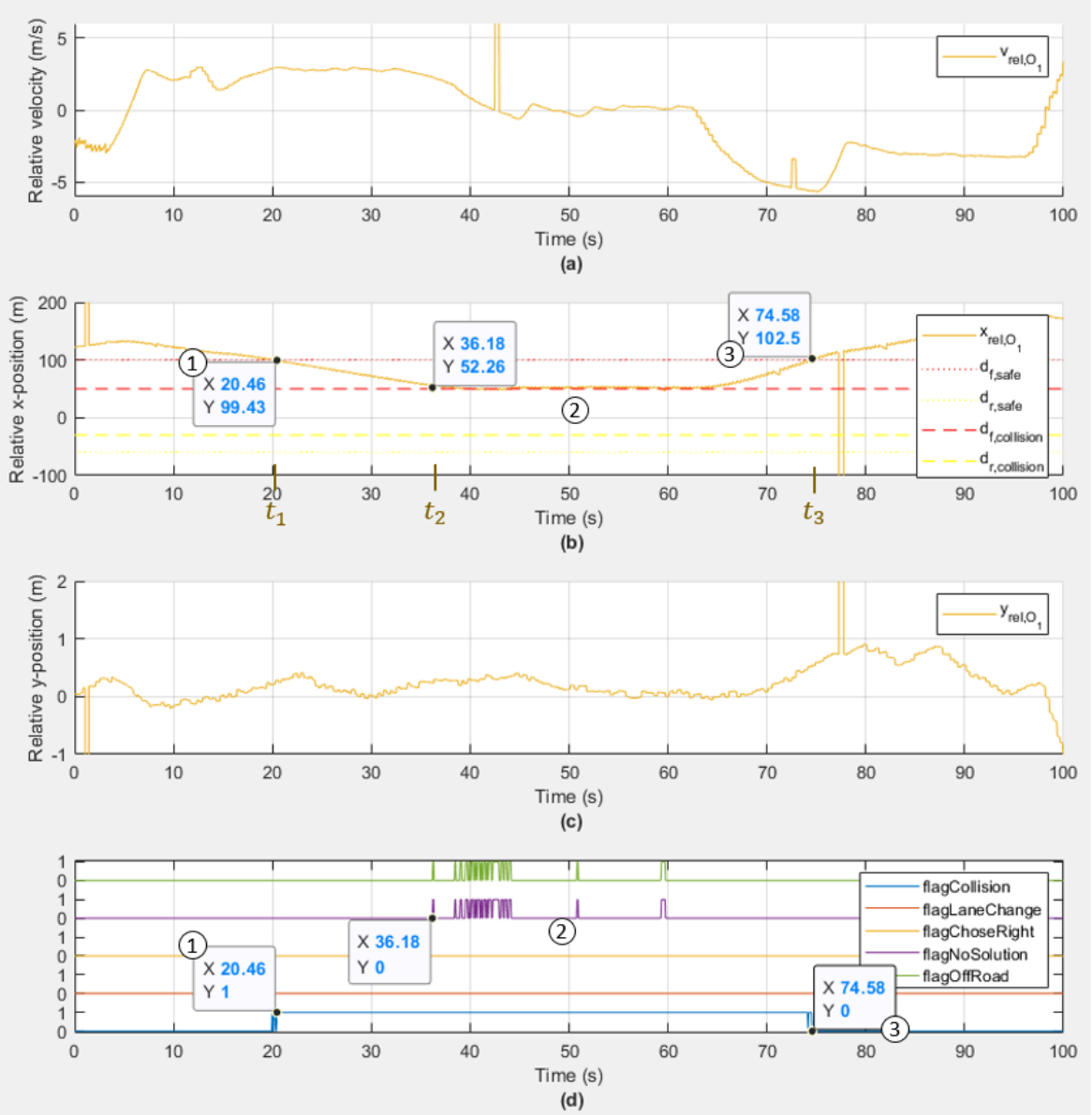


Figure 5.16: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis in critical case. The position errors of O_1 are due to errors in the communication blocks. [Scenario 2]

5.2.3 Scenario 3: Overtaking one front obstacle

The scenario 3 consists in overtaking the front obstacle O_1 on a 2-lane road. The overtaking maneuver is split into three phases: lane changing to the left behind the front obstacle at t_2, passing the obstacle on its left lane at t_3, and lane changing back to the right in front of the obstacle at t_4, as depicted in Figure 5.17. In this scenario, two intervals are defined for the first lane change – straight front I_1 and left I_2 in Figure 5.18a, and three intervals for the right lane change maneuver – straight front I_1, right rear O_1 for I_2, and right front O_1 for I_3 in Figure 5.18b. This scenario is tested with O_1 at a constant velocity lower than our wish velocity $v_{O_1,t_0} < v_{wish}$. As soon as O_1 violates the safety zone of the ego vehicle at t_1, i.e. $x_{rel,O_1} < d_{f,safe}$, the motion planner analyzes the situation in order to make a decision that will provide the best compromised trajectory. In this scenario, the best decision is to overtake O_1, which is initiated at t_2. The first lane change ends at t_3. Subsequently, the passing phase follows until O_1 is considered as a rear right obstacle. The final lane change starts at t_4, as soon as the decision-maker finds a collision-free ego velocity profile for a lane change to the right. At t_5, the ego vehicle is back on its initial lane in front of O_1 with a sufficient rear collision distance $d_{r,collision}$.

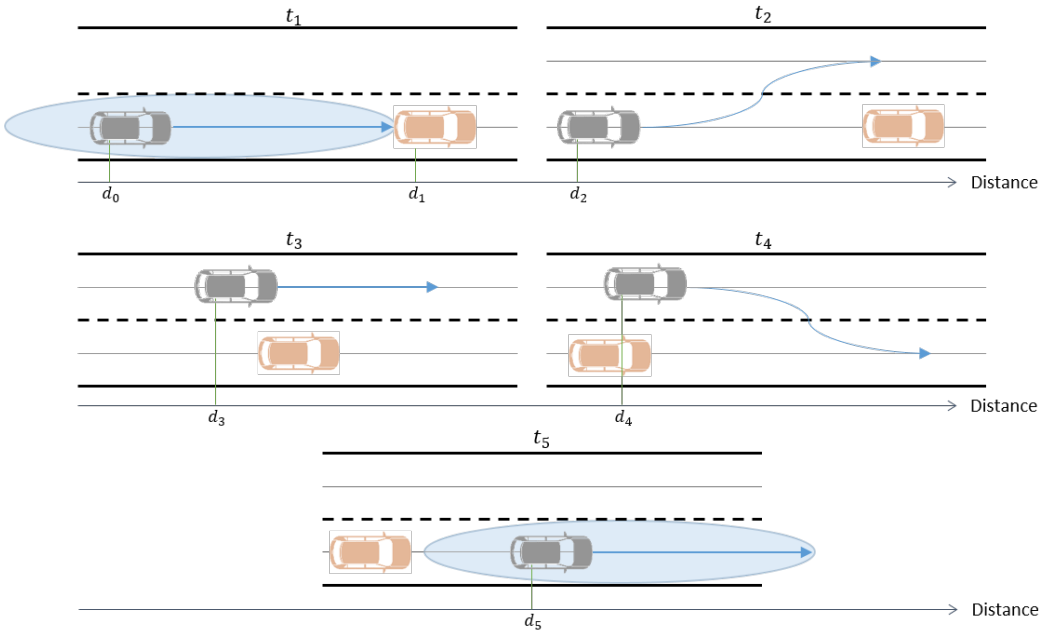
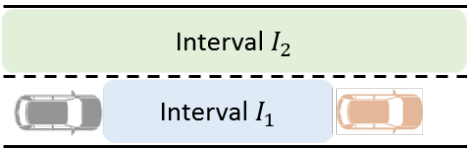
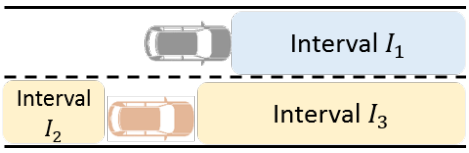


Figure 5.17: Illustration of scenario 3.



(a) First lane change.



(b) Second lane change.

Figure 5.18: Intervals representation for scenario 3.

When O_1 is detected below the safety distance $d_{f,safe}$ at $t_1 = 12.6$ s (data point 1 in Figure 5.20b), the motion planner sends a `True` collision flag value (data point 1 in Figure 5.20d). Then, a new reference trajectory χ_{ref} is calculated. As the left lane is free of obstacle, the best trajectory in order to respect the 3 decision criteria is a left lane change maneuver, as indicated by the lane change flag in Figure 5.20d (data point 1) and illustrated with χ_{ref} in Figure 5.19a (data area 1). This decision is made immediately at $t_2 = t_1 = 12.6$ s. The selected reference velocity for this maneuver is set at $v_{ref} = 11$ m/s (data point 1 in Figure 5.19b). One notices that v_{ref} is not equal to v_{wish}. This is due to the fact that the fuzzy logic returns a flat evaluation function as explained in Figure 5.22 and the Simulated Annealing architecture returns the first global minimum encountered. Indeed, the FIS presents a lot of constant warning areas. This means that the evaluation function will also presents these constant areas which are similar to local minima. Moreover, the evaluation of two solutions in the constant areas will return the same value. This can be an inconvenient when searching for the globally optimal solution. A set of close trajectories (i.e. for which the variables present little variation from one trajectory to another) is evaluated as near-optima. However, the non-uniqueness of a solution is still acceptable in our experiments for autonomous driving, because only the obstacle safety category is considered here. This particularity will also influence the neighborhood search in the simulated algorithm method: (i) it is important that the neighborhood search is large at high temperature, so that candidate solutions from different constant areas can be tested, and (ii) it appears useless to continue the SA search in a small neighborhood over a constant warning areas. The integration of the category 5 – Reference wishes from Table 3.1 (p. 55) in the decision-maker can solve this issue.

Besides, this first reference lane change is well followed (Figure 5.19a), and the collision distances are maintained as long as O_1 is on the ego lane (Figures 5.19a and 5.20b).

When the lane change is over at $t_3 = 20.1$ s (data points 2 in Figures 5.20c and d), the reference behavior consists in maintaining v_{wish} in the straight direction (data area 2 in Figure 5.19a and data point 2 in Figure 5.19b). Then, once the ego vehicle has passed O_1 (data points 3 in Figure 5.20b and c), the `right choice` flag is activated at $t_4 = 39.78$ s (data points 3 in Figure 5.20d), which means that the motion planner is triggered to return a reference trajectory to navigate in the right lane. As $v_{ego} > v_{O_1}$, the best maneuver is a rightward lane change ahead of O_1. This is decided and initiated at time t_4. Indeed, the motion planner has enough rear distance ($x_{rel,O_1} > d_{r,collision}$) to maneuver and navigate back on the right lane at $t_5 = 48.96$ s (data points in Figures5.20b, c and d). This second lane change is also well followed (data area 4 in Figure 5.19a).

Moreover, the lateral dynamics limits are respected for both the acceleration and angle in Figures 5.19c and d. In Figure 5.19d, one observes the left deviation of the steering angle when the ego vehicle is on the right lane (data areas 5 and 6) and the right deviation when the ego vehicle is on the left lane (data area 7) due to the road camber.

In Figure 5.21, we notice that the calculation times for the 2 maneuvers (data points 1 for left lane changing and 2 for right lane changing) are below 50 ms, which proves the real-time effectiveness of our algorithm.

To conclude, we notice that the fuzzy logic tuning is important to return the expected behavior (e.g. driver wishes). Our evaluation function, as represented in Figure 4.13 (p. 104), presents flat areas. This means that several different trajectories have the same evaluation function. According to the architecture of the Simulated Annealing algorithm, the output

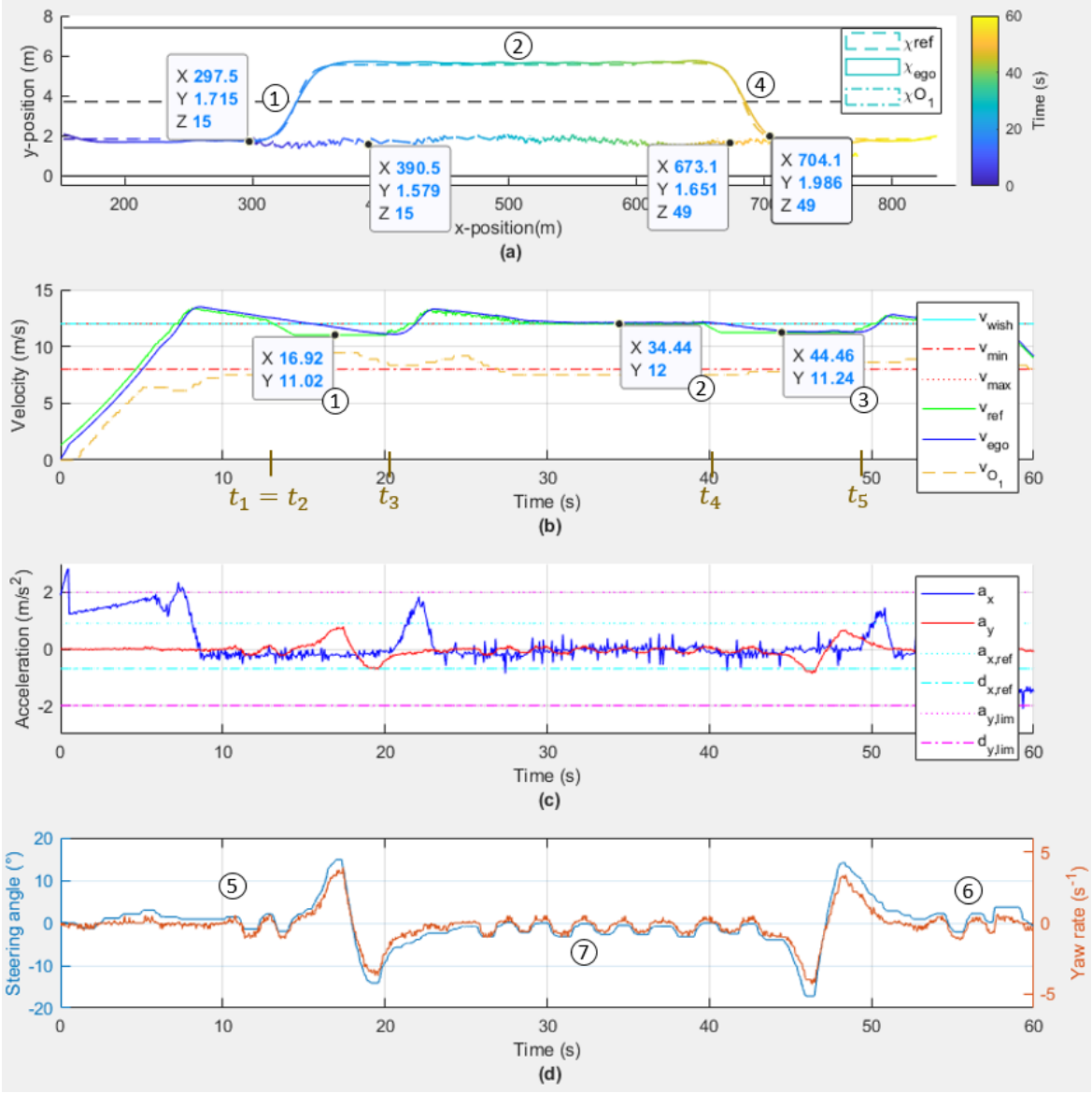


Figure 5.19: Trajectory, velocity and dynamics analysis. [Scenario 3]

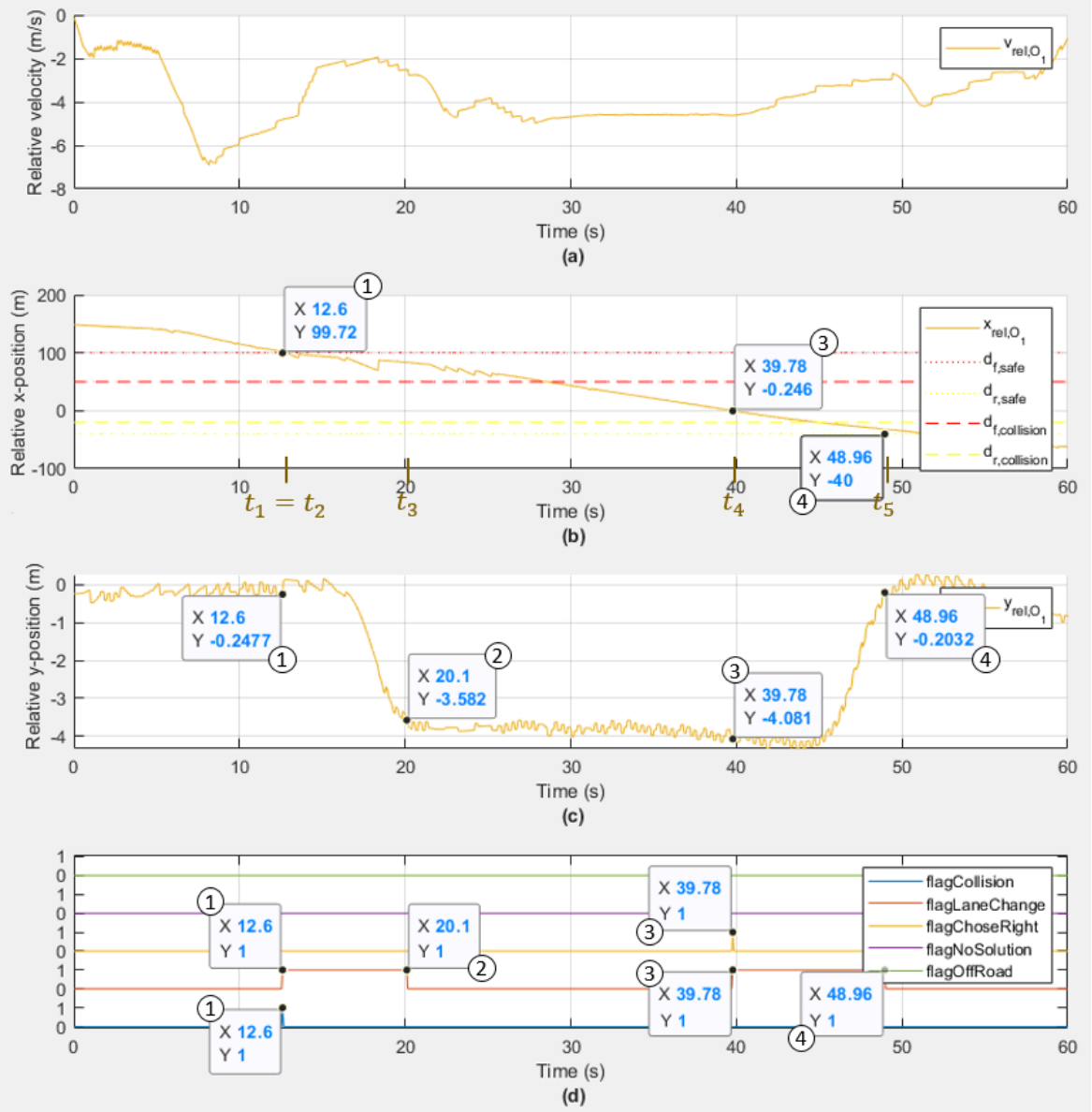


Figure 5.20: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis. [Scenario 3]

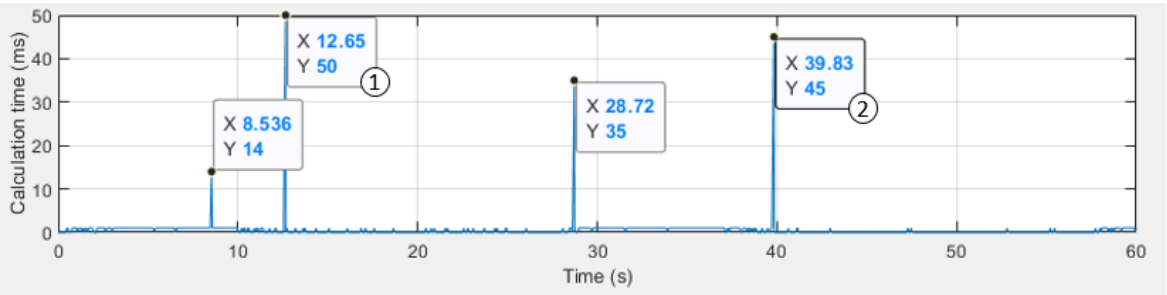


Figure 5.21: Calculation time. The peaks 1 and 2 correspond to the time step at which the motion planner is triggered. The two other peaks are outliers in the motion planner proceedings. [Scenario 3]

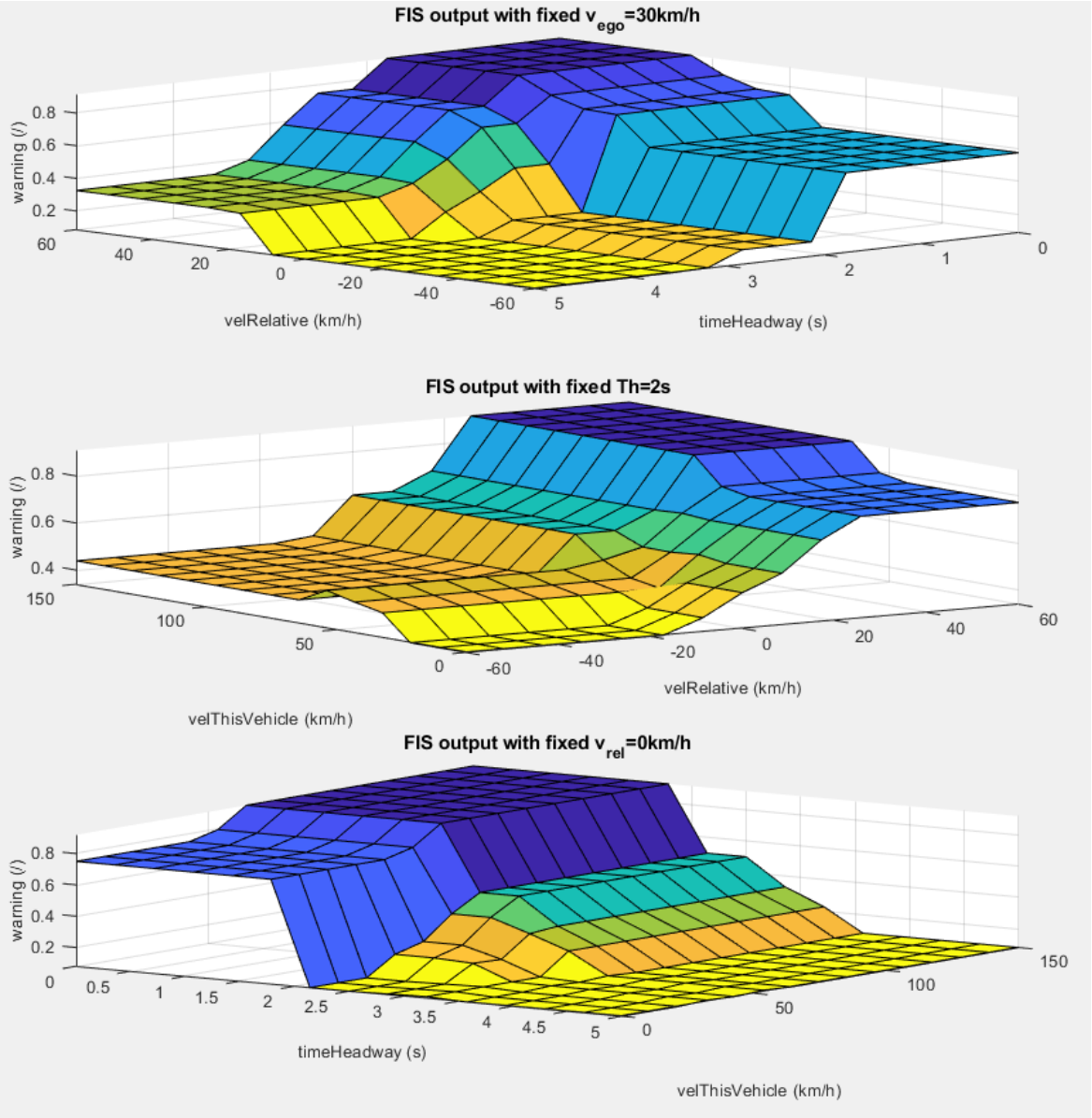


Figure 5.22: FIS output membership degree for the 3 criteria. Top: variations of relative velocity and time headway with fixed $v_{ego} = 30$ km/h. Middle: variations of relative velocity and ego velocity with fixed $T_h = 2$ s. Bottom: variations of time headway and ego velocity with fixed $v_{rel} = 0$ km/h. The darker and higher warning is, the riskier it is, and vice versa.

value is the first trajectory with the best evaluation function. Thus, it might be interesting to test our algorithm when considering the category 5 – Reference wishes to verify that the best velocity corresponds to the wish velocity v_{wish} when this latter features a global optimum too.

5.2.4 Scenario 4: Overtaking two front obstacles

A second obstacle O_2 to overtake is added to the previous scenario, see illustration in Figure 5.23. Both obstacles are navigating on the right lane at a velocity below the wish velocity v_{wish}. The two obstacles are close to each other (less than 50 m), and do not allow a merging of the ego vehicle between them. One of the hypotheses of our algorithm is to consider only the closest front obstacle in the ego lane. Thus, in this scenario, our algorithm only sees O_1 at t_1 with the interval representation in Figure 5.24a. When the ego vehicle navigates to the left lane at t_2, it sees O_2 in front of O_1, both in the right lane with the intervals in Figure 5.24b. The motion planner then calculates at t_3 whether a left lane changing maneuver $I_{1\to3}$ is safe, or if it should pass the 2 obstacles at once, staying in I_1. The latter is the safest solution trajectory as O_1 and O_2 are too close to each other. At t_4, the ego vehicle initiates a right lane change $I_{1\to4}$ in Figure 5.24c, which ends at t_5 when the ego vehicle is in front of O_2 on the right lane.

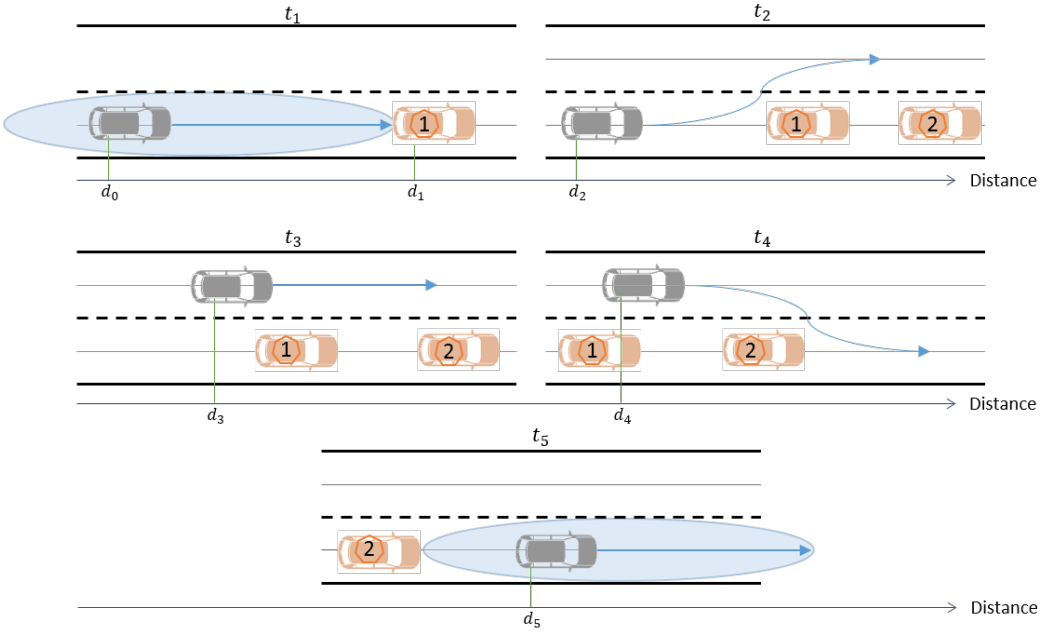
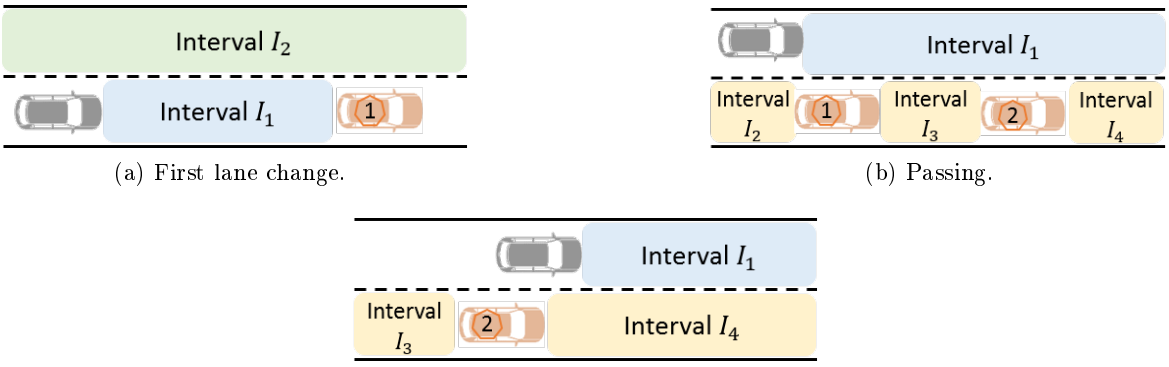


Figure 5.23: Illustration of scenario 4.

O_1 is detected by the LIDAR and O_2 communicates its positions and velocity (see subsection 5.1.2.2 for more details). As the LIDAR obstacle detection is stable within a distance of 50 m, the initial v_{wish} is set to 8 m/s (data point 1 in Figure 5.25b), $d_{f,safe}$ = 50 m, $d_{f,collision}$ = 20 m, $d_{r,safe}$ = 37.5 m, $d_{r,collision}$ = 15 m in order to have time for manually selecting O_1 in DeTroLi algorithm. O_1 is thus detected with x_{rel,O_1} = 58.1 m (data point 0 in Figure 5.26b). When x_{rel,O_1} is below $d_{f,safe}$ at t_1 = 50.28 s (data points 1 in Figures 5.26b, c



(a) First lane change.

(b) Passing.

(c) Second lane change.

Figure 5.24: Intervals representation for scenario 4.

and d), the motion planner calculates a new reference trajectory χ_{ref}, and decides a left lane change with $v_{ref} = 8.359$ m/s (data area 2 in Figure 5.25a and data point 2 in Figure 5.25b). This lane change is very aggressive with only 30 m to find the sigmoid distance delay c. We observe that χ_{ego} has an overshoot of 40 cm and 4.5 m longitudinal error, which corresponds to 720 ms time delay.

At the end of left lane change $t_3 = 55.44$ s (data points 3 in Figures 5.26c and d), we decide to change v_{wish} at 11 m/s (data point 3 Figure 5.25b).

When the ego vehicle reaches the left lane, O_2 becomes visible. The motion planner determines that there is not enough space to return to the right lane and discards the right lane change maneuver. In Figure 5.26d, the `right choice` flag is not activated.

When the ego vehicle has passed O_2 at $t_4 = 71.1$ s (data point 4 in Figure 5.26b), the motion planner searches and finds a trajectory to go back to the right lane in front of O_2 (data points 4 in Figures 5.26d), with $v_{ref} = 11.28$ m/s (data areas 4 in Figures 5.25a and b). The right lane change ends at $t_5 = 80.58$ s with $x_{rel,O_2} = 48.76$ m bigger than $d_{r,collision}$ (data points 5 in Figures 5.26b, c and d).

Finally, the longitudinal and lateral accelerations and angle constraints for both safety and comfort are respected (Figures 5.25c and d).

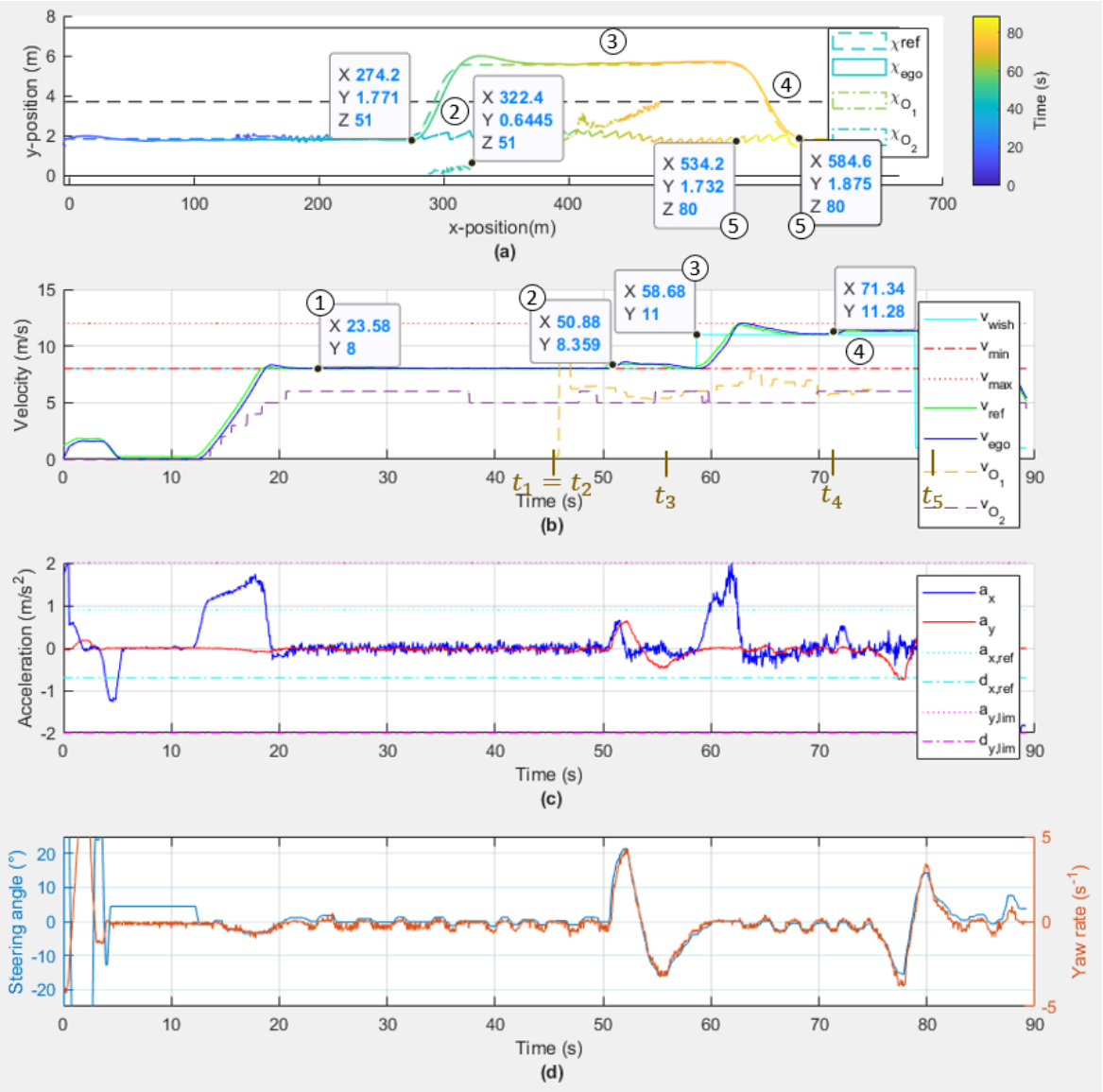


Figure 5.25: Trajectory, velocity and dynamics analysis. [Scenario 4]

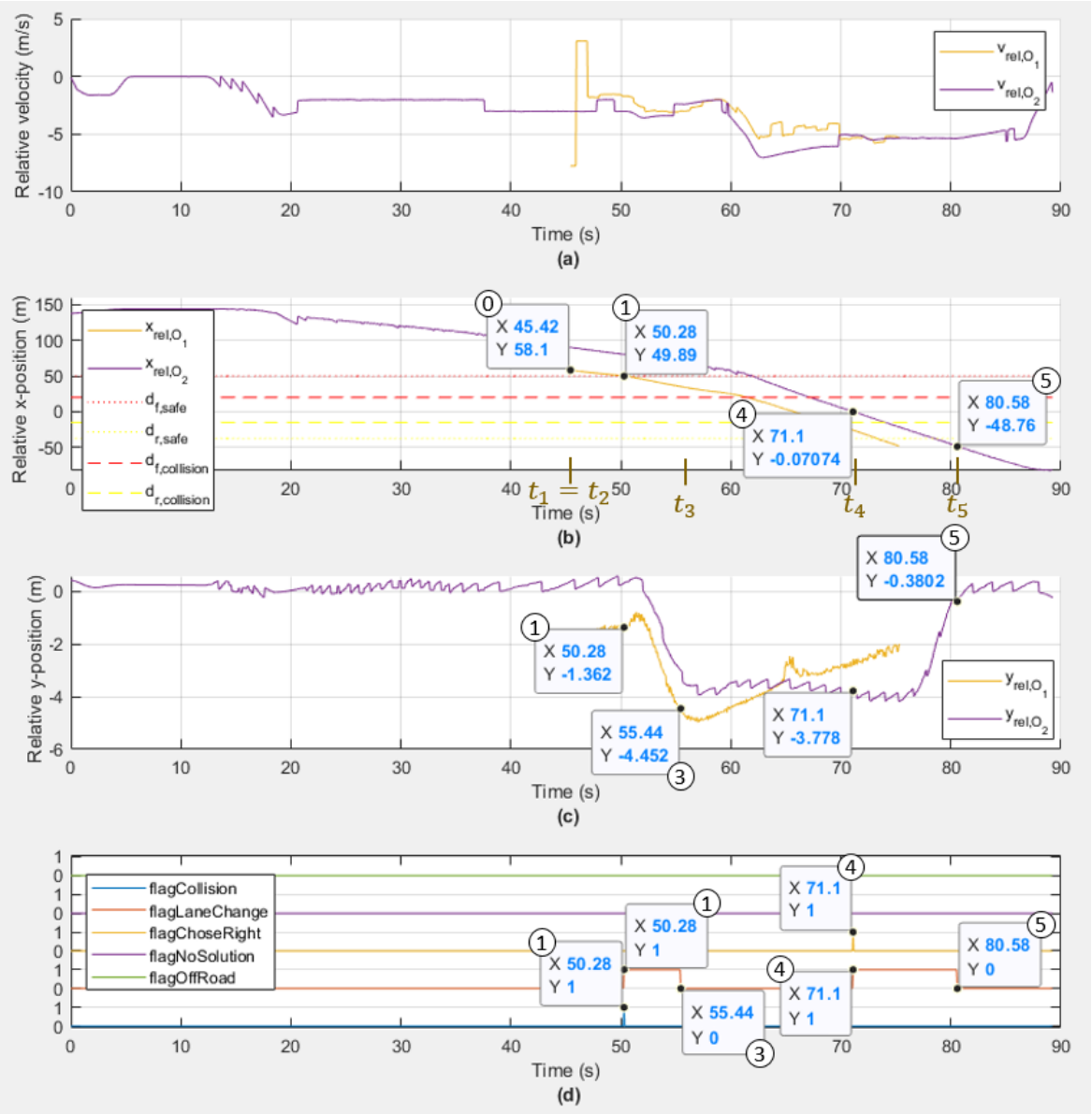


Figure 5.26: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis. [Scenario 4]

5.2.5 Scenario 5: Overtaking in front and behind a left obstacle

For a lane changing with two or more obstacles, the ego vehicle has to consider both the dynamics of the obstacle to overtake and the one of the obstacles in the passing lane. To illustrate how our motion planner works in this situation, two scenarios are presented with two obstacles (obstacle O_1 is the one initially in the front ego lane, obstacle O_2 is the one initially on the passing lane):

- If O_2 is behind the ego vehicle and its relative velocity v_{rel,O_2} and longitudinal distance x_{rel,O_2} are sufficient for the ego vehicle to pass in front of it and behind O_1, see subsection 5.2.5.1;
- If O_2 is behind the ego vehicle but v_{rel,O_2} and x_{rel,O_2} are not sufficient for the ego vehicle to pass in front of it, or if O_2 is in front but it blocks a lane changing for the ego vehicle because v_{rel,O_2} and x_{rel,O_2} are not advantageous in comparison to stay in the ego lane. In this situation, the ego vehicle has to wait for sufficient v_{rel,O_2} and x_{rel,O_2} in order to make a lane change and pass O_1 behind O_2, see subsection 5.2.5.2.

For both scenario the evolution space is decomposed into three intervals: straight front I_1, left front O_2 for I_2 and left rear O_2 for I_3 for the first lane change, see Figure 5.27a; straight front I_1, right front O_1 for I_2 and right rear O_1 for I_3 for the second lane change to go to the mostright lane after overtaking O_1, see Figure 5.27b.

(a) First lane change.

(b) Second lane change.

Figure 5.27: Intervals representation for scenario 5.

As explained in scenario 4, we set $d_{f,safe} = 50$ m, $d_{f,collision} = 20$ m, $d_{r,safe} = 37.5$ m, $d_{r,collision} = 15$ m. The scenarios are tested with $v_{wish} = 11$ m/s and $v_{O_1,t_0} \approx v_{O_2,t_0} \approx 8$ m/s.

5.2.5.1 Scenario 5.1: Overtaking a slower ego front vehicle with a slower rear left vehicle

This first maneuver is depicted in Figure 5.28. O_1 is in front ego lane and O_2 is rear on the left lane. When O_1 enters the safety area at t_1, the motion planner is triggered. As $v_{wish} > v_{O_1}$ and $v_{wish} > v_{O_2}$, the best decision is to pass O_1 in front of O_2, whose relative velocity and position guarantee rear safety. This decision is initiated at t_2. The lane change ends at t_3. The ego vehicle passes O_1 until it can go back into the right lane in front of O_1, as the relative velocity and distance with O_2 are sufficient not to be in collision.

O_1 is detected at $t_1 = 18.3$ s and is already below $d_{f,safe}$ (data points 1 in Figures 5.30b and d). The motion planner is triggered. However, v_{O_1} is calculated equal to 0 m/s (data points 1 in Figures 5.29b and 5.30a) and $x_{rel,O_1} < d_{f,safe}$. Consequently, there is no solution: no velocity profile for lane following, nor place for lane changing; χ_{ref} is the default trajectory

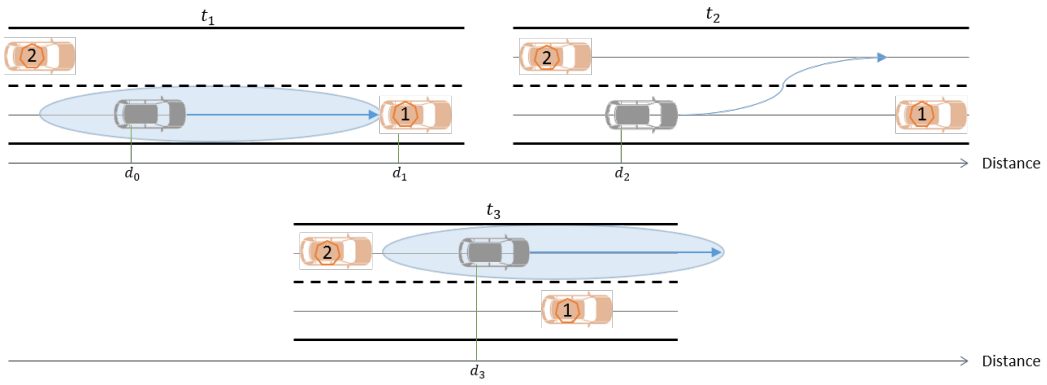


Figure 5.28: Illustration of scenario 5.1.

(data area 2 in Figures 5.29a and b) and the flag `offroad` is activated (data point 2 in Figure 5.30d). Nonetheless, as x_{rel,O_1} is still below $d_{f,safe}$, the flag collision is still activated (data areas 2 in Figures 5.30b and d). Thus, the motion planner is triggered every time-step to find the best χ_{ref}. For $t = [18.36, 18.78]$ s, χ_{ref} is staying on the ego lane. Indeed, v_{O_1} is over the maximum speed limit v_{max} (data area 3 in Figure 5.29b). Staying on the ego lane or left lane changing are equivalent.

At $t_2 = 18.84$ s, the motion planner finds a better evaluation function I_p for lane changing with $v_{ref} = 11.12$ m/s (data points 4 in Figures 5.29b and 5.30d). The ego vehicle overtakes O_1 in front of O_2 while maintaining the collision distances, respectively $d_{f,collision}$ and $d_{r,collision}$ (data area 4 in Figure 5.29a and data points 4 in Figures 5.30b and c). When on the left lane, $v_{ref} = v_{wish} = 11$ m/s (data point 0 in Figure 5.29b). When O_2 is behind the ego vehicle on its ego lane, it will no longer be analyzed by the motion planner according to *Assumption 6* (p. 94).

Once O_1 is overtaken, the motion planner suggests to go back into the right lane with $v_{ref} = 11.28$ m/s at $t_3 = 28.62$ s while maintaining $d_{r,safe}$ (data area and points 5 and 6 in Figures 5.29a and b and 5.30b and d).

Finally, the longitudinal and lateral accelerations and angle constraints for both safety and comfort are respected as shown in Figure 5.29c and d.

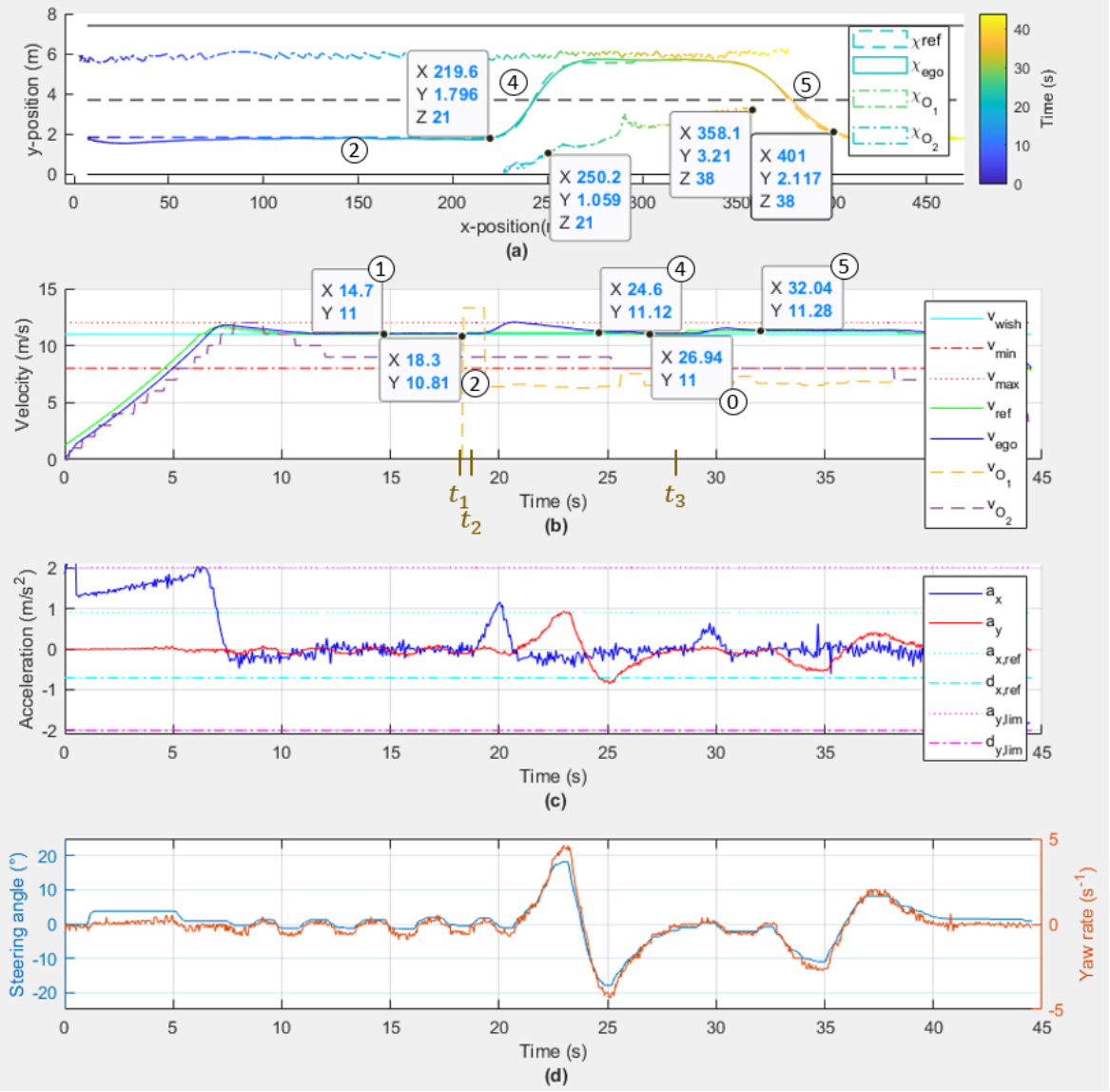


Figure 5.29: Trajectory, velocity and dynamics analysis. [Scenario 5-1]

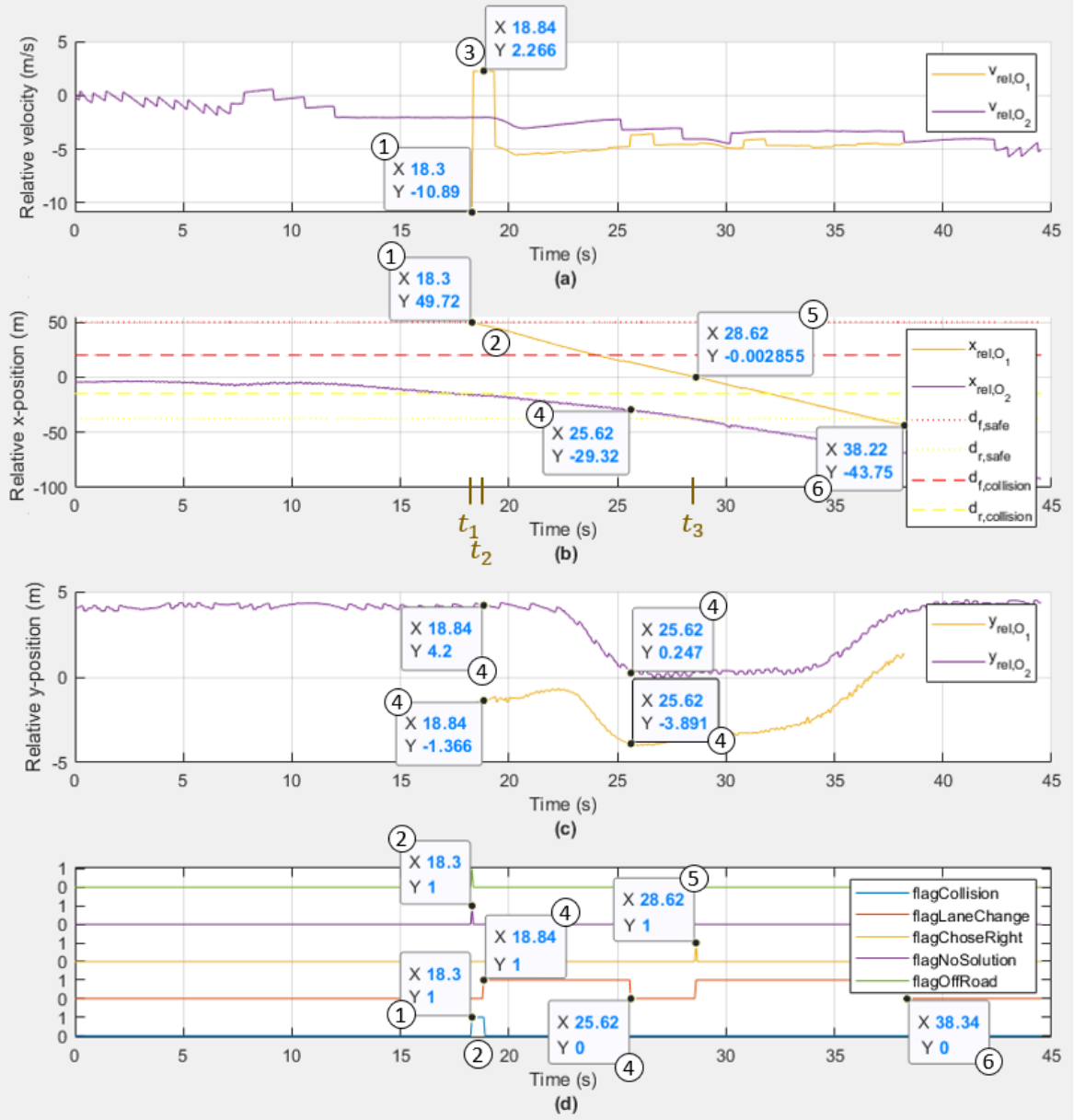


Figure 5.30: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis. [Scenario 5-1]

5.2.5.2 Scenario 5.2: Overtaking a slower ego front vehicle with a blocking left vehicle

This last scenario in Figure 5.31 is tested with O_1 in front ego lane and O_2 in front left lane behind O_1. The minimum speed limit v_{min} is set to 6 m/s as the vehicles are close to each other. When O_1 enters the safety area at t_1, the motion planner is triggered. The best decision is first to follow O_1. Indeed, O_2 is closer to the ego vehicle. The evaluation of any left lane change is worse than for staying trajectory. However, at t_3, O_2 accelerates to pass O_1. Thus, the left lane change is free of collision and has a better evaluation than staying behind O_1. Consequently, the motion planner decides to overtake O_1 behind O_2. The lane change ends at t_4. The ego vehicle passes O_1 and goes back into the right lane in front of O_1.

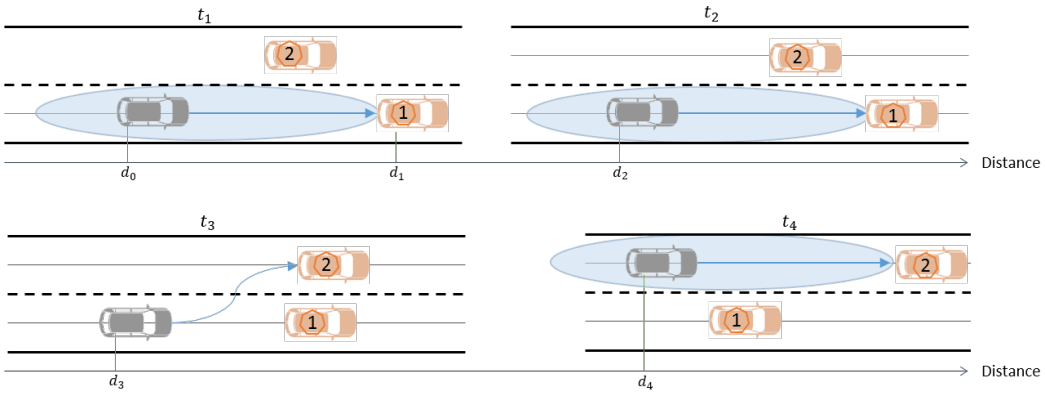


Figure 5.31: Illustration of scenario 5.2.

O_1 is below $d_{f,safe}$ at $t_1 = t_2 = 20.52$ s (data points 1 in Figures 5.33b and d). The motion planner is triggered. As $v_{O_2} \approx v_{O_1}$ and $x_{rel,O_1} > x_{rel,O_2}$, the left lane change maneuver has a lower evaluation function I_p than staying in the ego lane. However, the ego vehicle is still in collision with O_1 (data area 2 in Figure 5.33d). Consequently, the motion planner applies a car following behavior with O_1 and decreases v_{ref} to adapt to v_{O_1} (data area 3 in Figures 5.32b and 5.33a).

When O_2 increases its velocity at $t_3 = 38.34$ s, the left lane changing maneuver becomes safe and has a higher evaluation function I_p. The motion planner sends a reference left lane change with $v_{ref} = 8.246$ m/s (data area and points 4 in Figures 5.32a, b and 5.33d). We observe that χ_{ego} has an overshoot of 57 cm and 4.2 m longitudinal error, which corresponds to 660 ms time delay. This is due to the small lane change space behind O_1. One notices that the flag collision is activated when the lane changing is completed (data point 5 in Figure 5.33d). This is due to O_2 in the safety space of the ego vehicle (data points 5 in Figures 5.33b and c). Nonetheless, χ_{ref} is still a lane following, as O_2 is driving faster than the ego vehicle (data area 5 in Figure 5.32a).

Once O_1 is overtaken at $t_4 = 55.2$ s (data point 6 in Figure 5.33b), the motion planner suggests to go back into the right lane (data point 6 in Figure 5.33d). The right lane change starts at $t_5 = 55.74$ s with $v_{ref} = 10.58$ m/s (data area and points 7 in Figures 5.32a and b and 5.33d), and safely ends at $t_6 = 65.52$ s (data points 8 in Figures 5.33b and d).

Finally, the longitudinal and lateral accelerations and angle constraints for both safety and comfort are respected as shown in Figures 5.32c and d.

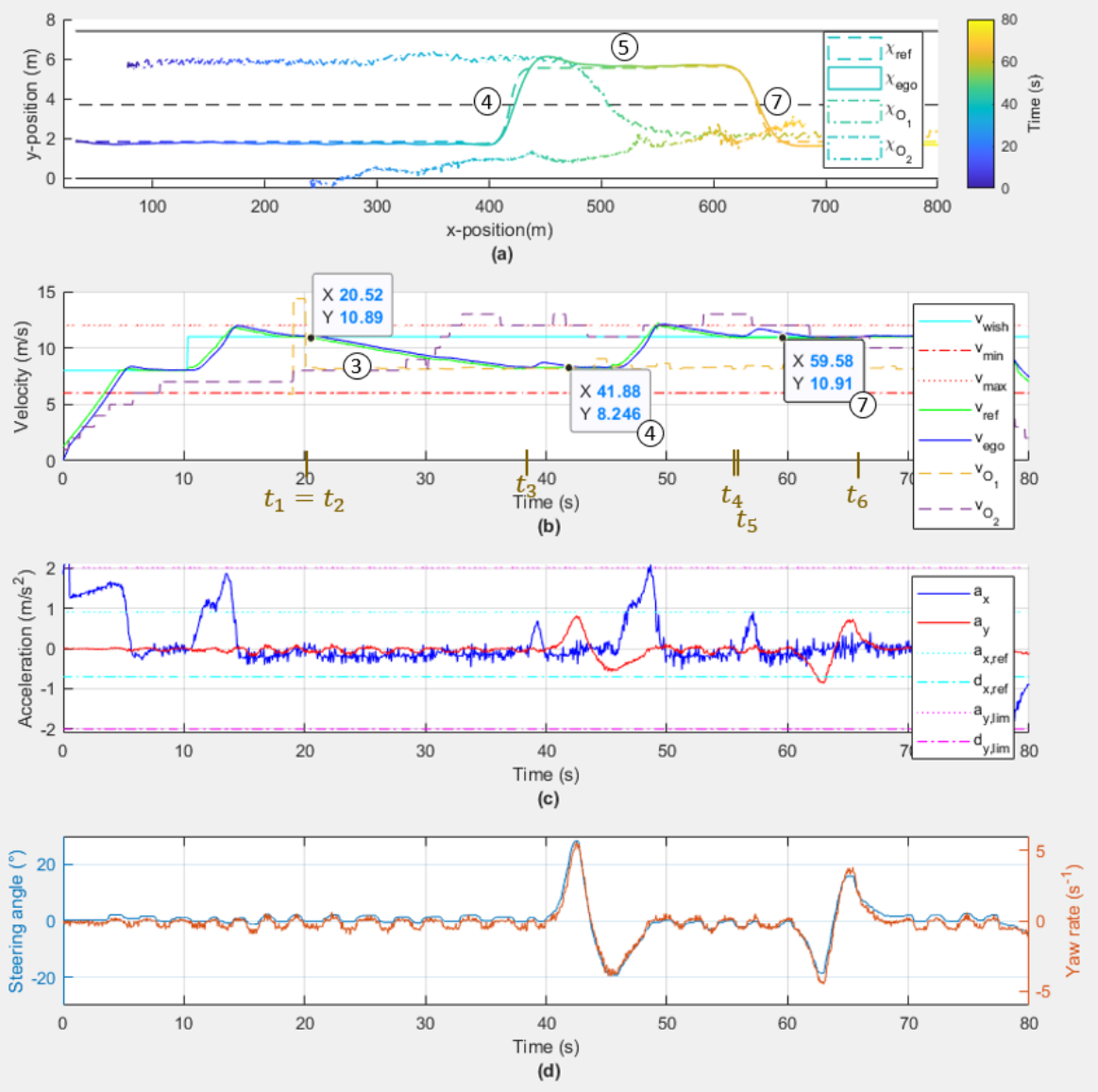


Figure 5.32: Trajectory, velocity and dynamics analysis. [Scenario 5-2]

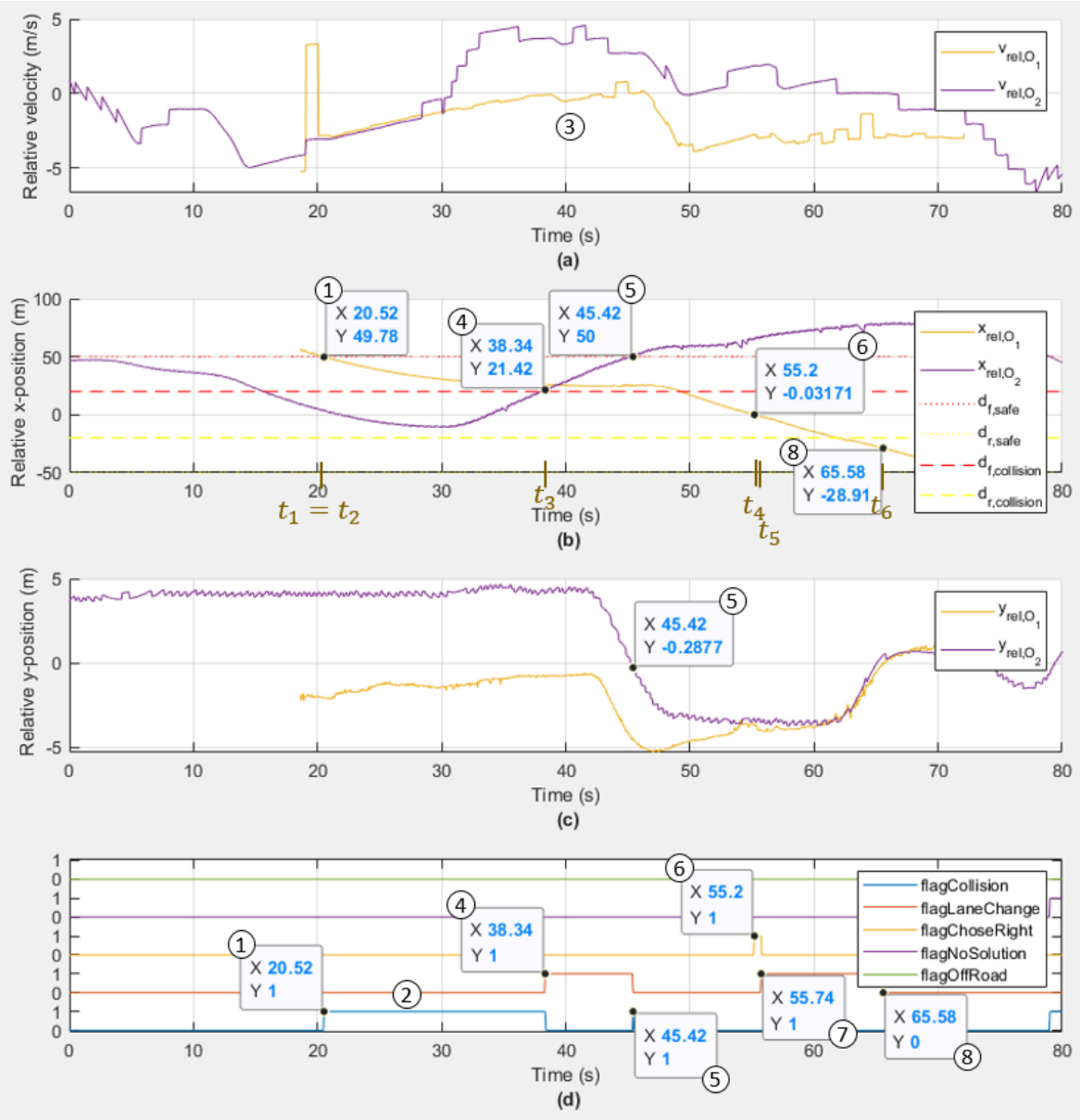


Figure 5.33: Relative longitudinal velocity, longitudinal and lateral positions and flag analysis. [Scenario 5-2]

5.2.6 Algorithm performances

In order to analyze the performances of the algorithm, we show in this subpart the convergence and time calculation of the Simulated Annealing method for all the scenarios, except scenario 1 (no planning).

Table 5.4 indicates the number of times where the motion planner has been triggered. For scenarios 3 and 4, the motion planner needs only one iteration to find a trajectory for both the left lane change and the right lane change. For scenario 5-1, the motion planner needs 10 iterations to find the appropriate maneuver. This is due to the small distance between the ego vehicle and O_1 for the left lane changing. Scenarios 2 and 5-2 have a higher number of iterations as the motion planner refers to a car following behavior with replanning at each time step.

Table 5.4: Number of triggering of the motion planner.

Scenario	2	3	4	5-1	5-2
Nb of triggering of the motion planner	838	2	2	10	303

The convergence iterations histogram for all the scenario is plotted in Figure 5.34. We also display the convergence of the evaluation function for scenarios 3, 4 and 5-1 in Figure 5.35. The maximum number of iteration for the Simulated Annealing was fixed at 200 for real-time integration. We notice that the algorithm converges in less than 200 iterations, which validates our algorithm tuning and our approach. With the chosen parameters, the convergence value at 5% is around 120 iterations. However, if there is a small variables search space, the convergence is faster as the corresponding evaluation function is probably within a flat area. This is the case at the end of the scenario 2, which leads to the smaller peak in Figure 5.34 around 70 iterations.

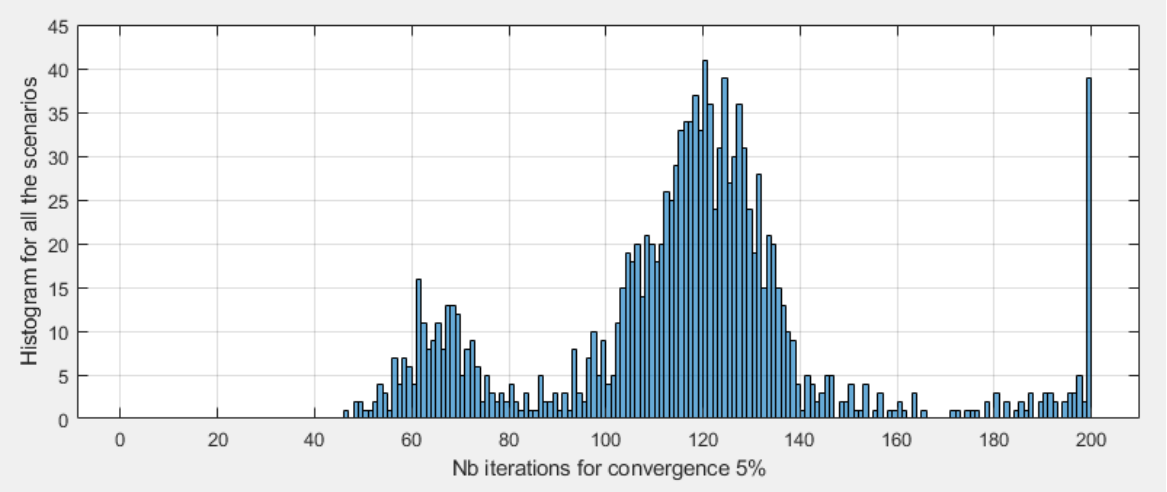
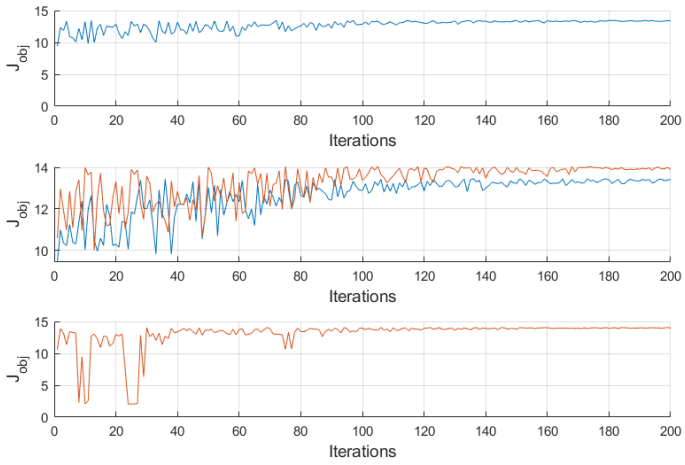
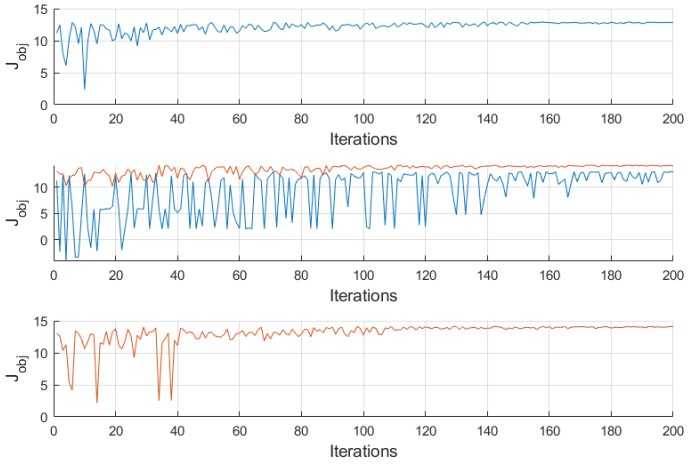


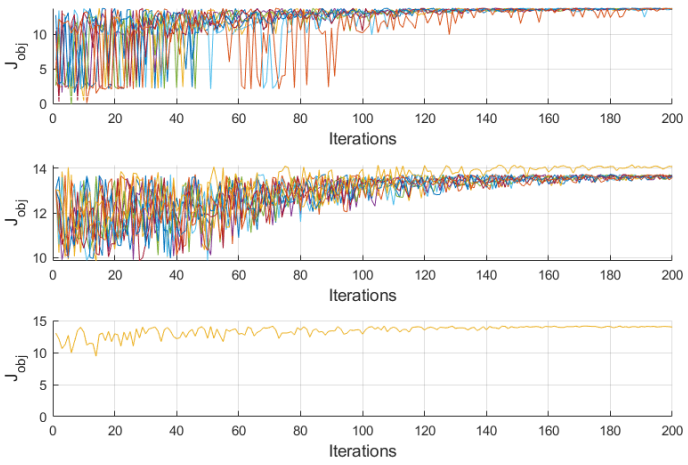
Figure 5.34: Histogram of the calculation time for all the scenarios.



(a) Convergence for scenario 3.



(b) Convergence for scenario 4.



(c) Convergence for scenario 5-1.

Figure 5.35: Convergence to the near-optimal value of the evaluation function $\mathcal{J}_{obj}$ for Simulated Annealing. For each scenario, the top, middle, and bottom figures respectively show the convergence for the left lane change, straight following, and right lane change.

5.3 Conclusion and future work

The experimental tests made it possible to demonstrate that the algorithm works as expected in all the described scenarios. The unified architecture of both the decision making and trajectory generation gives real-time computation for continuous optimization of the reference trajectory. Moreover, considering real obstacles localization with communication and perception algorithms shows the robustness of our algorithm to noisy inputs, which is promising to extend our work to real applications. Furthermore, even if the scenarios have been staged, the algorithm was not fixed on particular scenario values, and was able to adapt to the real behavior of the obstacles. The use of a simple dynamic model for the ego vehicle motion has also been validated with the use of a simple controller and a satisfying reference trajectory tracking in velocity and position (small overshoot and steady error).

However, the tests highlight some aspects of the methodology that could be blocking, in the sense of an overly conservative behavior for the implementation of the proposed algorithm in real life. The main changes to be made to the algorithm in future work are as follows:

- Having a continuous fuzzy inference system (FIS): The flat area of the evaluation function suggests that several different combinations of criteria will have the same risk indicator. This is not a problem for the motion planner itself, but it can lead to a non-effective optimization process. Testing a continuous fuzzy system could increase the efficacy of the optimization process and provide a more intuitive decision process.

- Relaxing *Assumptions 4* and *5* (p. 94) to discard a profile having a collision along the prediction horizon: The experimental results show that the collision test becomes too conservative with a large prediction horizon ($H_p > 5s$). On one hand, it may be possible to replan after a lane change completion for the front collision test (*Assumption 4*). On the other hand, the lane change back is far away from what can be observed in real life (*Assumption 5*). Besides, an anticipation horizon can be considered, which would be smaller than the prediction horizon and correspond to the minimum safe horizon for the ego vehicle to reduce or increase its velocity in order to stay safe.

- Relaxing *Assumption 6* (p. 94) not to consider the ego lane rear obstacles: In fact, the Driver Rules recommend to perceive and analyze its environment before acting. Therefore, the prediction of the rear obstacle has to be considered before making a decision.

- Relaxing *Assumption 10* (p. 119) freezing the lane change maneuver: A lane change maneuver might not be safe along the navigation, and the motion planner should be able to react to any surroundings change.

- The prediction horizon H_p should be both velocity- and distance-dependent as well as flexible with the maneuver. For example, a lane changing maneuver needs a higher prediction time than a lane following. In this case, $H_p = 5$ s for lane changing and $H_p = 1$ s for lane following would be sufficient.

- Considering the acceleration as a fourth variable to optimize in order to consider not constant acceleration for the maneuvers. This consideration will lead to more aggressive behaviors.

- Calculating the safety distances based on the ego and relative velocities to adapt the safety risk to the context. For instance, the safety distances at low velocities are smaller than the one at high velocities, similarly with a small relative velocity and a higher one.
- Considering more complex obstacles' prediction and testing the algorithm with uncertainties (see section 6.2 for more details).
- Parallelizing the algorithm to decrease the calculation time below 50 ms, and thus being able to consider more intervals.
- Dealing with traffic behavior to observe the integration of our ego vehicle in traffic. The first aspect is to test the impact of our decision in traffic and to notice the limit of good working for our algorithm. The second one is to test the interaction of many autonomous vehicles that use our motion planning solution. In particular, to verify the robustness of the decision-maker in case of aggressive or conservative fuzzy sets.

Chapter 6

Conclusion and Future Work

Contents

This last chapter summarizes and concludes the developments presented along this manuscript in section 6.1 and proposes future work and perspectives regarding our approach in section 6.2. Section 6.3 discusses more generally the gaps to be filled in motion planning by the next autonomous driving car generation.

6.1 Conclusions on our contributions

In this thesis, we proposed a motion planner algorithm for autonomous vehicles in a highway environment, integrating a unified decision-maker and trajectory generator able to handle multi-criteria consideration and offering a continuous optimization of a spatiotemporal evolution space. This manuscript details the different developments resulting from our research, from the study of the state of the art to the experimentations of our proposed strategy.

The study of the literature, presented in Chapter 2, has evidenced four problems in research on motion planning. The first one (i) is the use of separated or sequential architecture of the two studied motion planning functions –decision making and trajectory generation– either by first applying decision making to decide on a maneuver and then proceeding to motion generation to properly fit this maneuver, or vice versa. Treating these two tasks in a separate fashion may indeed lead to a final motion that is not consistent or optimal with respect to the evolution space. The second issue (ii) concerns the discrete handling of the navigation area. Indeed, if the discretization is too small, the evolution space may be truncated; conversely, if the discretization is too high, the calculation time may be too long. The third area of improvement (iii) is that trajectories are usually obtained by separately optimizing velocity and position. A spatial path solution is defined on a static decomposition of the space, the path is adjusted to the dynamic obstacles by choosing a velocity/acceleration profile, or reverse by defining the optimal velocity profile dynamically and then the static path. These first three aspects may harm the consistency of the motion planner and most often reduce the solution-space. Finally, the last point to be considered (iv) concerns the treatment of multi-criteria decision making without context. By defining the nature of a criterion in the sense of its value and its evolution in a context, it is necessary to provide the human passengers with an understandable reasoning on the contribution of each criterion, in particular regarding the context. For instance, at low velocities, a small time headway is acceptable, whereas it is not at high velocities.

In order to address the issues evidenced in Chapter 2, we first investigated a strategy to incorporate the nature of the criteria into multi-criteria decision making, and then developed a unified architecture for decision-maker and trajectory generator.

In Chapter 3, we focused on the decision making function. The main objectives were to propose a deterministic multi-criteria approach considering the uncertainties of perception, while avoiding an overly conservative vehicle's behavior. This presents two main issues: the heterogeneous criteria and the uncertain data. The proposed framework for decision making addresses both by combining the fuzzy logic approach and the Belief theory. Using the fuzzy logic to translate the specific criteria scales into a risk scale, which is common to all the criteria, we were able to consider heterogeneous criteria without loosing their nature, and to combine them into a human readable risk assessment, thus proposing a solution to the literature problem (iv). Moreover, the criteria combination follows a rule-base, which allows for a decision-reasoning that is easily understandable and traceable for humans, and also for the interpretation of the context before making the decision. Besides, tuning the fuzzy set now offers the possibility to adapt the risk scale to the passengers of the vehicle. As all the criteria may not have the same importance regarding the passenger' behavior, we also used a belief theory framework within the Fuzzy Dempster-Shafer Theory. The advantage of this framework is its ability to assign criteria to different categories corresponding to

different decision-states, such as safety or comfort. The categories are then considered as diverse sources of information to obtain the final decision. Thus, the combination of the categories returns two major information for decision making: the probability to be in the state and the belief in actually being in this state. Moreover, the consideration of uncertainties and trustworthiness becomes possible by using fuzzy logic and weighting the criteria and categories. Finally, the Fuzzy Dempster-Shafer reasoning avoids both the defuzzification process for fuzzy logic by using belief theory and the problem of fallacy of the excluded middle for belief theory by using fuzzy set and appropriate rule-base.

In Chapter 4, we first studied the problem of the sequential motion planning architecture for the literature problem (i), proposing a unified approach so that both functions of motion generation and decision making are optimized in a combined manner. To do so, we developed an all-in-one architecture, within which the decision making algorithm from Chapter 3 corresponds to the evaluation function used in the optimization approach for trajectory generation. At the end, this architecture returns a near-optimal reference trajectory to follow. The main advantage is thus the joint processing of decision making and trajectory generation resulting in a consistent motion planner. Another advantage of this architecture is its ability to work in a continuous trajectory solution space, which eludes the issue of trajectory discretization (literature problem (ii)). Secondly, we presented a deterministic and continuous characterization of the evolution space, considering it as a set of intervals. We introduced the Non-Collision Nominal Intervals (NCNI), which consist in a collision-free space delimited by road lanes and spaces between obstacles. The NCNI contain all the spatiotemporal motions allowed for the ego vehicle in the evolution space along the prediction horizon. The high-level maneuvers are carried out to move from one interval to another. The position and velocity prediction of obstacles and the dynamically feasible motions of the ego vehicle characterize the spatiotemporal intervals. By optimizing a trajectory inside these intervals, we also avoid the decoupling issue in motion generation (literature problem (iii)). Lastly, in order to utilize the implicit evaluation function of the decision-maker developed in Chapter 3, we have opted for a metaheuristics approach with a Simulated Annealing algorithm, that allow us to find a near-optimal solution in a finite time. Furthermore, it offers the possibility to adjust the calculation time according to the encountered situation, while guaranteeing to return an acceptable solution.

In Chapter 5, we successfully demonstrated the effectiveness of our approach implemented in an autonomous prototype vehicle. The motion planning algorithm has been integrated with the controller and perception modules. The tested scenarios highlighted the real-time and robustness performances of our work with two manually-driven obstacles.

6.2 Future work regarding our contributions

The research solutions presented in this thesis have shown encouraging results. The following future work will enrich the reflection carried out to effectively bring these ideas into a fully autonomous vehicle application.

Extending the highway environment Even if we tested some merging use cases with the 2-obstacle scenarios, future work will consist in addressing highway ramp-entrances and

exits. This environment requires a socially cooperative driving: either the ego vehicle enters the highway and merges into the current traffic, or the ego vehicle is in traffic and adjusts its own behavior to facilitate the merging of the incoming vehicles. Besides, considering this road context will also change the nominal conditions of driving, otherwise the ego vehicle will be confronted with a blockage action. For instance, the relative safe distances or ego velocity for merging can be reduced. Both situations of merging refer to the ability to plan adequate courtesy behaviors, which may lead to the creation of a new courtesy category in our work.

Dealing with uncertainty From a theoretical standpoint, the current development of our Fuzzy Dempster-Shafer approach can accommodate trustworthiness. Additional research is needed to build a Fuzzy Dempster-Shafer reasoning which also considers uncertainty as input of the fuzzy logic. Moreover, considering several obstacles' predictions is necessary, as the predicted behaviors are only accurate for a short time horizon. One of the direction could be the addition of weights on the criteria according to the probability and dangerousness of the predicted obstacle's behavior, as it has been done with the data trustworthiness.

Improving the fuzzy relationships The definition of the fuzzy sets and the associated rule-base is of major importance in our decision making process. We noticed the two following issues: (i) the consistency of the rule-base, and (ii) the constant warning areas in the evaluation function. The first improvement requires further research to make the rule-base coherent and consistent in order to prevent non-intuitive results and missing association of the fuzzy sets. The second improvement consists in studying the coherence and influence of the fuzzy set membership functions on the evaluation function. For example, a trapezoidal membership function will return a flat evaluation function, which results in an area with local minima. The non-uniqueness of a solution is acceptable for motion planning in the case of autonomous vehicles, but it influences the optimization process, which can become inefficient.

Making explicit the decision-maker We developed a deterministic decision-maker. In the proposed method, each criterion is indirectly associated to risk levels, the combination of all the criteria of a category returns a local warning for the category, and the combination of all the categories returns a global risk assessment. Thus, it can be cumbersome to clearly identify the criteria leading to an unsafe evaluation of the decision-maker. Consequently, improvements have to be done in order to propose an easily readable interface for humans to understand why a maneuver is evaluated as unsafe or safe.

Dealing with ethics behavior In order to adapt the behavior of the fuzzy inference systems to a human-in-the-loop behavior, learning techniques can also be considered for different driver profile datasets to form the fuzzy sets. This improvement is useful for at least two reasons: (i) reproducing a human behavior leads to a better acceptation in case of a mix human- and robot-driver, (ii) learning the fuzzy sets will consist in adapting the risk levels to the context, and thus integrating somehow a cognitive reasoning.

6.3 Future research directions in motion planning for autonomous vehicles

This section spotlights some critical and forthcoming issues of research in the wider scope of motion planning for autonomous vehicles in general.

6.3.1 Data management

The scene representation input in motion planning algorithms is characterized by data, their uncertainties and trustworthiness. The quality and quantity of these data are crucial to ensure safe decisions. For example, the authors in [Wang *et al.* 2017] propose a method to evaluate the appropriate number of naturalistic driving data to model a car following behavior. Some research has been conducted on data uncertainties and trustworthiness on applied methods, such as Kalman estimators [Yu *et al.* 2016, Xu *et al.* 2014], Markov processes [Ulbrich & Maurer 2015, Althoff & Mergel 2011], Monte Carlo simulation [Althoff & Mergel 2011], evidential theory [Yu *et al.* 2016], or interval arithmetic [Althoff & Dolan 2014]. Furthermore, the planning module must also ensure motion continuity to guarantee the safety of the vehicle even if data are missing, or with latency and unsynchronized data. In our thesis work, we considered the last available data. However, this solution can become critical with more complex traffic situation and less time to react.

6.3.2 Adaptive mobility

Adaptive mobility raises the question of the introduction of autonomous vehicles to a human environment. In fact, all the infrastructure and driving rules for transportation means are built from human models. Thus, it can be wise to ask whether it is reasonable to expect to model and to reproduce human behavior/reasoning on a robotic system whose environmental knowledge is not adapted in terms of perception and reasoning means. Three aspects of motion planning are especially subject to this question: safety, perception compensation, and route context.

Safety The safety considerations impose the need for intelligent enough algorithms to distinguish a permanent or temporary danger, such as the front vehicle suddenly stopping, or another one cutting in.

From a motion planning perspective, the ego vehicle is safe if it is not in conflict with its environment. However, this safety space highly depends on the safety capacity of robot driving, which can be different from a human one. As stated in [Althoff & Lösch 2016], longitudinal distance and speed controls are generally faster in autonomous driving, and so longitudinal safety space could be smaller. In contrast, lateral distance control is generally more stable with human drivers; thus, lateral safety space must be larger. Furthermore, this safety space has to be sufficient against the unexpected challenges of the environment (e.g. unexpected braking from leading vehicle, unforeseen road's dead end). The formal methods are thus used to mathematically prove safety properties [Pek *et al.* 2017a, Pek *et al.* 2017b] in any conditions including the most critical ones.

Moreover, at the moment, automakers are developing automated vehicles with human takeover and human-robot mixed driving. This implies a possibly shared decision making between automated vehicles and drivers. The first problem is retaining a stable and predictive decision in case the driver takes the control back [Merat *et al.* 2014]. The second one is that the decisions of the human driver and the machine may contradict each other [Da Lio *et al.* 2015]. The third safety consideration is of human drivers' behaviors towards robot drivers in the sense of respect, acceptability, and predictability [Verberne *et al.* 2012], or comfort [Elbanhawi *et al.* 2015].

Perception compensation One of the problems regarding perception is the constraint of a fixed point of view of the driving scene. However, when a human driver has to make a decision, he/she optimizes his/her perception by moving the ego vehicle to capture more information on a larger perceived environment. This perception compensation places the ego vehicle in a position to optimize the capacity of the sensors with a geometrically wide perception, as suggested in [Andersen *et al.* 2017] for city driving. This can be adapted to highway driving in order to anticipate an exit-ramp or a traffic jam, thus constraining the reference trajectory along the road lanes.

Route context The route planner is well studied with a trip scheduler. However, there are very few examples in the literature of integrating these data for decision and generation functions. In fact, route planning constrains the evolution space, especially on highways; e.g. a right-hand highway exit involves the ego vehicle to navigate to the rightmost lane. These requirements imply respecting constraints in actions, distances, and time to follow the instructions of the navigator.

6.3.3 Validation and evaluation

The evaluation of the algorithms must include a verification of behaviors, judgments, and responsibilities. This evaluation can be done qualitatively, as we did in Chapter 5, but motion planning research is missing a common evaluation framework in order to compare the algorithms to each other. In addition to the algorithmic performances, the validation of transition stability with different planners, the evaluation of motion planning methods on predefined use cases, the ethics dilemmas, and the topic of rules relaxation are important in the implementation of the motion planner.

Transition stability The variety of the encountered situations often requires the use of different combined motion planning algorithms. In this case, we need to know whether the resulting architecture is stable, reliable and robust, or whether it is necessary to add a high-level supervisor to validate the coherence of observations, decisions and actions. In [Broggi *et al.* 2013b], the authors propose a behavior paradigm analogous to the FSM decision to implement an architecture able to switch maneuver modes from manual to autonomous driving.

Evaluation The validation of motion planning algorithms mostly relies on extensive simulation and experimental testing results, where a human driver evaluates the action according to

a personal reference index, such as safety, smoothness, or operation time [Tehrani *et al.* 2015]. There is a lack of formal analysis and evaluation methods to hierarchically classify algorithms' performances. This would change thanks to the open source scenario library developed by [Althoff *et al.* 2017] to propose a benchmark for motion planning algorithms, or with the Key Performance Indicators (KPIs) proposed by [Quilbeuf *et al.* 2018].

Ethics As soon as robotic systems interact with human beings, questions of ethics in decision-makers arise. Reference [Goodall 2014] proposes an ethical vehicle deployment strategy in a hybrid rational and AI approach, especially for critical safety situations. The authors in [Thornton *et al.* 2017] show how to incorporate ethics into decisions based on ethical frameworks, such as deontology –as rules constraining the actions and navigation of the system, consequentialism –as cost-based construction for the objective function, or as morality to determine the different costs of the system's behaviors.

Algorithm relaxation The strict respect and relaxation of driving rules are key points to ensure safety in any case for an autonomous vehicle. Indeed, even hard constraints might become dangerous in critical cases [Broggi *et al.* 2014]. Reference [Morignot & Nashashibi 2013] introduces an ontology for traffic rules in unusual situations with knowledge-based inference engines to avoid blockage situations and to preserve safety with lane crossing or excess speeds.

Appendix A

Lateral acceleration limits calculation

A.1 Second-order time derivative of the sigmoid function

The t-parameterized equation of the sigmoid function, with $x(t)$ the abscissa value and $y(t)$ the ordinate value, is given in Equation A.1:

$$y(t) = \frac{b}{1 + e^{-\lambda(x(t)-c)}}, \tag{A.1}$$

where b is the shift value, λ is the sigmoid parameter, and c an abscissa delay of the inflection point.

In our problem, b, λ and c are constant values, and $x \mapsto x(t)$ is at least a twice differentiable function over $\mathbb{R}$. Consequently, y is obviously at least a twice differentiable function of the time variable t.

The first-order time derivative of x is written $\frac{\mathrm{d}x}{\mathrm{d}t} \triangleq v_x$.

The second-order time derivative of x is written $\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = \frac{\mathrm{d}v_x}{\mathrm{d}t} \triangleq a_x$.

The first-order and second-order sigmoid derivatives with respect to x are respectively expressed in Equations A.2 and A.3:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{-b(-\lambda)e^{-\lambda(x-c)}}{(1 + e^{-\lambda(x-c)})^2} = \lambda y(1 - \frac{y}{b}) = \frac{\lambda}{b} y(b - y). \tag{A.2}$$

$$\begin{aligned} \frac{\mathrm{d}^2y}{\mathrm{d}x^2} &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\frac{\lambda}{b}y(b-y)}{\mathrm{d}x}\right) \\ &= \frac{\lambda}{b}\left[\frac{\mathrm{d}y}{\mathrm{d}x}(b-y) - y\frac{\mathrm{d}y}{\mathrm{d}x}\right] \\ &= \frac{\lambda}{b}\left[\frac{\lambda}{b}y(b-y)^2 - \frac{\lambda}{b}y^2(b-y)\right] \\ &= \frac{\lambda^2}{b^2}y(b-y)(b-2y). \end{aligned} \tag{A.3}$$

The first-order time derivative of the sigmoid function represented in Equation A.1 is calculated using the chain rule in Equation A.4:

$$\begin{aligned} \frac{\mathrm{d}y}{\mathrm{d}t} &= \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t} \\ &= \frac{\mathrm{d}y}{\mathrm{d}x}v_x \\ &= \frac{\lambda}{b}y(b-y)v_x. \end{aligned} \tag{A.4}$$

The second-order time derivative of the sigmoid function based on Equation A.4 is calculated using the chain rule in Equation A.5:

$$\begin{aligned}\frac{\mathrm{d}^2y}{\mathrm{d}t^2} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)\\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t}\right)\\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}^2x}{\mathrm{d}t^2}\\ &= \frac{\mathrm{d}^2y}{\mathrm{d}t\mathrm{d}x}v_x + \frac{\mathrm{d}y}{\mathrm{d}x}a_x.\end{aligned} \tag{A.5}$$

Using the Schwarz's Theorem, we express $\frac{\mathrm{d}^2y}{\mathrm{d}t\mathrm{d}x}$ in Equation A.6 :

$$\begin{aligned}\frac{\mathrm{d}^2y}{\mathrm{d}t\mathrm{d}x} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)\\ &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)\\ &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t}\right)\\ &= \frac{\mathrm{d}^2y}{\mathrm{d}x^2}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}^2x}{\mathrm{d}x\mathrm{d}t}\\ &= \frac{\mathrm{d}^2y}{\mathrm{d}x^2}v_x + \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}^2x}{\mathrm{d}x\mathrm{d}t}.\end{aligned} \tag{A.6}$$

Using the Schwarz's Theorem, we express $\frac{\mathrm{d}^2x}{\mathrm{d}x\mathrm{d}t}$ in Equation A.7 :

$$\frac{\mathrm{d}^2x}{\mathrm{d}x\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}x}{\mathrm{d}x}\right) = \frac{\mathrm{d}}{\mathrm{d}t}(1) = 0. \tag{A.7}$$

Thus, Equation A.5 becomes Equation A.8:

$$\begin{aligned}\frac{\mathrm{d}^2y}{\mathrm{d}t^2} &= \frac{\mathrm{d}^2y}{\mathrm{d}t\mathrm{d}x}v_x + \frac{\mathrm{d}y}{\mathrm{d}x}a_x\\ &= \left[\frac{\mathrm{d}^2y}{\mathrm{d}x^2}v_x + \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}^2x}{\mathrm{d}x\mathrm{d}t}\right]v_x + \frac{\mathrm{d}y}{\mathrm{d}x}a_x\\ &= \frac{\mathrm{d}^2y}{\mathrm{d}x^2}v_x^2 + \frac{\mathrm{d}y}{\mathrm{d}x}a_x\\ &= \frac{\lambda^2}{b^2}y(b-y)(b-2y)v_x^2 + \frac{\lambda}{b}y(b-y)a_x.\end{aligned} \tag{A.8}$$

Finally, one has:

$$\boxed{\frac{\mathrm{d}^2y}{\mathrm{d}t^2} = \frac{a_x\lambda}{b}y(b-y) + \frac{v_x^2\lambda^2}{b^2}y(b-y)(b-2y)}. \tag{A.9}$$

A.2 Resolution of the second-order time derivative constraints

We want to find the bounds values of λ in order to respect the maximum lateral acceleration condition in Equation 4.9 (p. 102), as recalled in Equation A.10:

$$\frac{a_x\lambda}{b}y(b-y)+\frac{v_x^2\lambda^2}{b^2}y(b-y)(b-2y)\leq a_{y,\max} \tag{A.10}$$

We write:

$$\begin{aligned} f(y)&=\frac{a_x\lambda}{b}y(b-y)+\frac{v_x^2\lambda^2}{b^2}y(b-y)(b-2y),\\ f_1(y)&=\frac{a_x\lambda}{b}y(b-y),\\ f_2(y)&=\frac{v_x^2\lambda^2}{b^2}y(b-y)(b-2y), \end{aligned} \tag{A.11}$$

with $y\in[0,b]$, by definition of b in Equation A.1.

Equation A.10 is then equivalently reformulated as Equation A.12:

$$f_1(y)+f_2(y)\leq a_{y,\max} \tag{A.12}$$

Then, the two terms $f_1(y)$ and $f_2(y)$ of Equation A.12 are bounded separately.

- Bounds of function f_1: we study the variations of $f_1(y), y\in[0,b]$ in Table A.1.

Table A.1: Variation table of function f_1.

y	0		$\frac{b}{2}$		b
$f_1'(y)$		+	0	−	
$f_1(y)$	0	↗	$f_1(y=\frac{b}{2})$	↘	0

The maximum value of f_1 is for $y=\frac{b}{2}$: $f_1(y=\frac{b}{2})=\frac{a_x b}{4}\lambda$.
To conclude, we have Equation A.13:

$$\forall y\in[0,b],\quad 0\leq f_1(y)\leq\frac{a_x b}{4}\lambda \tag{A.13}$$

- Bounds of function f_2: we study the variations of $f_2(y), y\in[0,b]$ in Table A.2.

Table A.2: Variation table of function f_2.

y	0		$y_1=\frac{3-\sqrt{3}}{6}b$		$y_2=\frac{3+\sqrt{3}}{6}b$		b
$f_2'(y)$		+	0	−	0	+	
$f_2(y)$	0	↗	$f_2(y_1)$	↘	$f_2(y_2)$	↗	0

The maximum value of f_2 is obtained for $y=\frac{3-\sqrt{3}}{6}b$: $f_2(y=\frac{3-\sqrt{3}}{6}b)=\frac{\sqrt{3}}{18}v_x^2 b\lambda^2$.

To conclude, we have Equation A.14:

$$\forall y \in [0, b], \quad f_2(y) \leq \frac{\sqrt{3}}{18} v_x^2 b \lambda^2. \tag{A.14}$$

At the end, we obtain an upper bound for function f in Equation A.15:

$$\boxed{\forall y \in [0, b], \quad f(y) \leq \frac{a_x b}{4} \lambda + \frac{\sqrt{3}}{18} v_x^2 b \lambda^2}. \tag{A.15}$$

Consequently, we choose λ in order to respect Equation A.12:

$$\frac{a_x b}{4} \lambda + \frac{\sqrt{3}}{18} v_x^2 b \lambda^2 \leq a_{y,\max} \tag{A.16}$$

Let g be the function defined for $\lambda \in \mathbb{R}^+$ as:

$$g(\lambda) = \frac{\sqrt{3}}{18} v_x^2 b \lambda^2 + \frac{a_x b}{4} \lambda - a_{y,\max}. \tag{A.17}$$

Equation A.16 is equivalent to Equation A.18:

$$g(\lambda) \leq 0 \tag{A.18}$$

The variation table of g is given in Table A.3.

Table A.3: Variation table of function g

λ	0		λ_2		$+\infty$
$g(\lambda)$	$-a_{y,max}$	$-$	0	$+$	

with $a_{y,max} \geq 0$, $\lambda_2 = \frac{\frac{-a_x b}{4} + \sqrt{\Delta}}{\frac{\sqrt{3}}{9} v_x^2 b}$, and $\Delta = \frac{a_x^2 b^2}{16} + \frac{2\sqrt{3}}{9} v_x^2 b a_{y,\max} > 0$.

Finally, the maximum value for λ that fulfills the condition of Equation 4.9 is given in Equation A.19:

$$\boxed{\lambda_{\max} = \frac{-a_x b/4 + \sqrt{\Delta}}{\frac{\sqrt{3}}{9} v_x^2 b}, \quad \Delta = \frac{a_x^2 b^2}{16} + \frac{2\sqrt{3}}{9} v_x^2 b a_{y,\max}.} \tag{A.19}$$

Appendix B

Résumés en français

B.1 Chapitre 1 : Introduction

Le premier chapitre introductif présente le contexte et l'évolution des véhicules automatisés vers le véhicule autonome. Trois phases de recherche et développement se distinguent dans le temps. Une première phase de recherche s'est manifestée de manière isolée dans différents instituts des années 1920 jusqu'aux années 1980. Une deuxième phase de consortiums a été mise en place ensuite afin de pousser la recherche du véhicule autonome vers des systèmes et environnements plus complexes. De nombreuses expérimentations ont alors montré le potentiel de l'automatisation des véhicules. Depuis les années 2000, une troisième phase de développement des véhicules autonomes intéresse fortement les laboratoires de recherche et l'industrie automobile, et voit également émerger de nouveaux acteurs de services.

Le véhicule autonome soulève de nombreuses questions, dont celle portée par le travail de doctorat présenté dans ce manuscrit, à savoir la planification de mouvements d'un véhicule autonome à travers la conception d'une architecture unifiée de prise de décision et de génération de trajectoires. Les contributions portent sur quatre points de recherche : une revue de littérature orientée sur la planification sur autoroute, une nouvelle architecture de prise de décision et de génération de mouvements, une génération de manœuvre dans un espace spatio-temporel continu, et un cadre de décision adaptatif.

B.2 Chapitre 2 : Etat de l'Art

Le chapitre d'état de l'art détaille les considérations spécifiques à la recherche d'un planificateur de mouvement dans un environnement autoroutier. La terminologie propre à la planification de mouvement, les différents niveaux de planification ainsi que les spécificités et contraintes inhérentes au contexte autoroutier sont présentés.

Une revue de littérature est également proposée dans ce chapitre. Les algorithmes des fonctions de prise de décision, génération de mouvement et déformation sont analysés vis-à-vis de leur application à un environnement autoroutier. Trois axes sont étudiés : l'utilisation de l'algorithme, la nature des sorties qu'il renvoie et son orientation mathématique. Les algorithmes existants sont alors rassemblés en 6 familles. A l'issue de l'état de l'art, un tableau de comparaison est proposé de manière à mettre en évidence l'algorithme le mieux adapté au problème défini.

Pour finir, les conclusions de ce travail bibliographique font ressortir le besoin d'une méthode de planification de mouvements explicable et déterministe (réplicabilité), intuitive pour l'humain (acceptabilité), cohérente avec l'environnement d'application (adaptabilité et faisabilité en temps réel), et contextuelle. Ce travail de recherche se positionne donc sur une

réflexion de l'architecture de prise de décision multi-critères contextuelle et génération de mouvement dans un espace de navigation continu et spatio-temporel.

B.3 Chapitre 3 : Prise de décision

Ce chapitre traite du problème de prise de décision pour les véhicules autonomes sur autoroute. L'objectif est de sélectionner une trajectoire de référence sur un horizon prédictif. La trajectoire sélectionnée est obtenue à partir d'une optimisation multi-critères. Ces critères peuvent par exemple correspondre à la sécurité avec les obstacles, au respect des règles de conduite, au confort des passagers, à la consommation énergétique, etc.

Ce travail introduit un nouveau cadre de décision proposant une évaluation de risque. L'architecture est présentée en trois niveaux. A partir des mesures capteurs et de la prédiction de l'environnement sur un horizon de prédiction, pour chaque trajectoire à évaluer, on obtient les valeurs des critères. Le premier niveau utilise une approche par logique floue pour agréger ces critères de décision hétérogènes et incertains. En sortie de ce niveau, chaque trajectoire candidate se voit attribuer un niveau d'avertissement local. La théorie des croyances est utilisée au deuxième niveau pour combiner les niveaux d'avertissement locaux en les interprétant comme différentes sources de risque pour la prise de décision. A l'issue de ce niveau, chaque trajectoire candidate est associé à un intervalle de risque. Enfin, le troisième niveau consiste à évaluer l'intervalle de risque afin de définir la trajectoire la moins risquée comme trajectoire de référence. Des résultats en simulation sont finalement exposés dans ce chapitre pour illustrer le raisonnement appliqué.

B.4 Chapitre 4 : Génération de trajectoires

Le chapitre 4 considère la fonction de génération de mouvement et de manière plus globale l'architecture des deux fonctions de génération et de prise de décision.

Une nouvelle architecture est proposée, unifiant les deux fonctions de planification étudiées. La prise de décision établie dans le chapitre précédent est considérée comme la fonction coût d'une trajectoire sigmoïde dont les paramètres sont à optimiser dans la fonction de génération de trajectoires.

Le deuxième aspect de ce chapitre concerne l'espace de recherche pour la génération de trajectoire. Celui-ci est considéré comme un ensemble d'intervalles spatio-temporels décrivant ainsi un espace de navigation continu entre les obstacles le long de l'horizon de prédiction. Le passage d'un intervalle à un autre constitue une manœuvre du véhicule. Les intervalles sont ensuite caractérisés en vitesse et en distance selon un critère de non collision en condition nominale de conduite. Le processus d'optimisation s'applique au sein de ces intervalles pour le choix du profil de vitesse, du point de changement de voie et de la pente du changement de voie, paramètres de la courbe sigmoïde. Cela évite ainsi les problèmes de discrétisation et de découplage de trajectoire mis en évidence dans l'état de l'art. L'utilisation d'un algorithme de recuit simulé est enfin choisi pour permettre l'optimisation d'une fonction non-explicite et sans connaissance a priori de ses propriétés mathématiques. Des résultats en simulation sont également exposés dans ce chapitre pour illustrer le fonctionnement de l'algorithme.

B.5 Chapitre 5 : Expérimentations et résultats

Le chapitre 5 montre les expérimentations implémentées sur véhicule réel. Celles-ci se sont déroulées sur la piste de vitesse du site de Versailles-Satory. Le véhicule test est instrumenté par une localisation centrale inertielle et GPS RTK, un module de communication Ultra Haute Fréquence (UHF) et une ceinture de 5 LIDARs pour la détection d'obstacles. La cartographie est prédéfinie. Deux véhicules obstacles complètent les moyens expérimentaux. Le premier, disposant également d'une centrale inertielle et GPS RTK, communique sa position par communication UHF. Le second obstacle est perçu par les LIDARs. Les informations de localisation et de perception des obstacles sont envoyées au planificateur qui décide de la trajectoire de référence à suivre par le contrôleur. Ce dernier se compose d'un contrôleur proportionnel-intégral sur la consigne de vitesse, et d'un contrôleur proportionnel sur la consigne de position, en considérant l'angle de déviation.

Cinq scénarios ont été effectués pour ces expérimentations. Pour chacun des scénarios, le suivi de la trajectoire de référence, les limites de la dynamique du véhicule et les critères de sécurité de distance aux obstacles sont étudiés. L'analyse des résultats montrent la reproductibilité de l'algorithme, une trajectoire de référence réalisable et correctement suivie (faible dépassement et retard), ainsi que le respect des limites dynamiques du véhicules et des distances de sécurité aux obstacles. De plus, l'algorithme est temps réel avec un temps de calcul de 60 ms avec le paramétrage proposé.

B.6 Chapitre 6 : Conclusions et perspectives

Ce dernier chapitre fait état des conclusions du travail de thèse mené ainsi que des perspectives à développer. Des conclusions et perspectives plus générales sur le problème de planification sont également proposées.

Bibliography

[Aarts & Van Laarhoven 1985] E. H. Aarts et P. J. Van Laarhoven. *Statistical cooling: A general approach to combinatorial optimization problems.* Philips J. Res., vol. 40, no. 4, pages 193–226, 1985. (p. 99)

[Ali *et al.* 2013] M. Ali, A. Gray, Y. Gao, J. K. Hedrick et F. Borrelli. *Multi-objective collision avoidance.* In ASME Dynamic Systems and Control Conference, 2013. (p. 51)

[Altché & De La Fortelle 2017] F. Altché et A. De La Fortelle. *Partitioning of the free space-time for on-road navigation of autonomous ground vehicles.* In IEEE Annual Conf. on Decision and Control (CDC), 2017. (p. 26, 46, 85, 86 and 89)

[Altché *et al.* 2017] F. Altché, P. Polack et A. de La Fortelle. *High-speed trajectory planning for autonomous vehicles using a simple dynamic model.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2017. (p. 33 and 47)

[Althoff & Dolan 2014] M. Althoff et J. M. Dolan. *Online verification of automated road vehicles using reachability analysis.* IEEE Trans. on Robotics, vol. 30, no. 4, pages 903–918, 2014. (p. 161)

[Althoff & Lösch 2016] M. Althoff et R. Lösch. *Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance?* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2016. (p. 161)

[Althoff & Mergel 2011] M. Althoff et A. Mergel. *Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars.* IEEE Trans. on Intelligent Transportation Systems, vol. 12, no. 4, pages 1237–1247, 2011. (p. 161)

[Althoff *et al.* 2017] M. Althoff, M. Koschi et S. Manzinger. *CommonRoad: Composable benchmarks for motion planning on roads.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 163)

[Andersen *et al.* 2017] H. Andersen, W. Schwarting, F. Naser, Y. H. Eng, M. H. Ang, D. Rus et J. Alonso-Mora. *Trajectory optimization for autonomous overtaking with visibility maximization.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2017. (p. 162)

[Arbitmann *et al.* 2012] M. Arbitmann, U. Stählin, M. Schorn et R. Isermann. *Method and device for performing a collision avoidance maneuver*, Juin 26 2012. US Patent 8,209,090. (p. 32 and 46)

[Ardelt *et al.* 2010] M. Ardelt, P. Waldmann, F. Homm et N. Kaempchen. *Strategic decision-making process in advanced driver assistance systems.* IFAC Proceedings Volumes, vol. 43, no. 7, pages 566–571, 2010. (p. 25, 35, 39, 46 and 47)

[Aurenhammer 1991] F. Aurenhammer. *Voronoi diagrams – a survey of a fundamental geometric data structure.* ACM Computing Surveys (CSUR), vol. 23, no. 3, pages 345–405, 1991. (p. 26)

[Axelsson 2017] J. Axelsson. *Safety in Vehicle Platooning: A Systematic Literature Review.* IEEE Trans. on Intelligent Transportation Systems, vol. 18, no. 5, pages 1033–1045, 2017. (p. 19)

[Baass 1984] K. Baass. *Use of clothoid templates in highway design.* In Transportation Forum, volume 1, pages 47–52, 1984. (p. 31)

[Bahram *et al.* 2014] M. Bahram, A. Wolf, M. Aeberhard et D. Wollherr. *A prediction-based reactive driving strategy for highly automated driving function on freeways.* In IEEE Intelligent Vehicles Symposium (IV), 2014. (p. 27, 28, 40, 46 and 47)

[Balal *et al.* 2016] E. Balal, R. L. Cheu et T. Sarkodie-Gyan. *A binary decision model for discretionary lane changing move based on fuzzy inference system.* Transportation Research Part C: Emerging Technologies, vol. 67, pages 47–61, 2016. (p. 38, 47 and 51)

[Baluja *et al.* 1997] S. Baluja, R. Sukthankar et J. Hancock. *Prototyping intelligent vehicle modules using evolutionary algorithms.* In Evolutionary algorithms in engineering applications, pages 241–257. Springer, 1997. (p. 37 and 47)

[Bautin *et al.* 2010] A. Bautin, L. Martinez-Gomez et T. Fraichard. *Inevitable collision states: a probabilistic perspective.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2010. (p. 39 and 47)

[Bender *et al.* 2015] P. Bender, Ö. S. Tas, J. Ziegler et C. Stiller. *The combinatorial aspect of motion planning: Maneuver variants in structured environments.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 26 and 46)

[Bevly *et al.* 2016] D. Bevly, X. Cao, M. Gordon, G. Ozbilgin, D. Kari, B. Nelson, J. Woodruff, M. Barth, C. Murray, A. Kurt*et al.* *Lane change and merge maneuvers for connected and automated vehicles: A survey.* IEEE Trans. on Intelligent Vehicles, vol. 1, no. 1, pages 105–120, 2016. (p. 20)

[Bojarski *et al.* 2016] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang*et al.* *End to end learning for self-driving cars.* arXiv preprint arXiv:1604.07316, 2016. (p. 38 and 47)

[Borenstein & Koren 1991] J. Borenstein et Y. Koren. *The vector field histogram-fast obstacle avoidance for mobile robots.* IEEE Trans. on Robotics and Automation, vol. 7, no. 3, pages 278–288, 1991. (p. 25 and 30)

[Boroujeni *et al.* 2017] Z. Boroujeni, D. Goehring, F. Ulbrich, D. Neumann et R. Rojas. *Flexible unit A-star trajectory planning for autonomous vehicles on structured road maps.* In IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), 2017. (p. 28 and 46)

[Bounini *et al.* 2017] F. Bounini, D. Gingras, H. Pollart et D. Gruyer. *Modified artificial potential field method for online path planning applications.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 28, 29, 30 and 46)

[Broggi *et al.* 1999] A. Broggi, M. Bertozzi, A. Fascioli, C. Bianco et A. Piazzi. *The ARGO autonomous vehicle's vision and control systems.* Int. J. of Intelligent Control and Systems, vol. 3, no. 4, pages 409–441, 1999. (p. 3)

[Broggi *et al.* 2013a] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici et P. Versari. *Extensive tests of autonomous driving technologies.* IEEE Trans. on Intelligent Transportation Systems, vol. 14, no. 3, pages 1403–1415, 2013. (p. 3)

[Broggi *et al.* 2013b] A. Broggi, S. Debattisti, M. Panciroli et P. P. Porta. *Moving from analog to digital driving.* In IEEE Intelligent Vehicles Symposium (IV), 2013. (p. 162)

[Broggi *et al.* 2014] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, M. Panciroli et A. Prioletti. *Proud-public road urban driverless test: Architecture and results.* In IEEE Intelligent Vehicles Symposium (IV), 2014. (p. 3, 16, 17, 27, 46 and 163)

[Cardoso *et al.* 2017] V. Cardoso, J. Oliveira, T. Teixeira, C. Badue, F. Mutz, T. Oliveira-Santos, L. Veronese et A. F. De Souza. *A Model-Predictive Motion Planner for the IARA autonomous car.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2017. (p. 33 and 47)

[Černỳ 1985] V. Černỳ. *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm.* Journal of optimization theory and applications, vol. 45, no. 1, pages 41–51, 1985. (p. 97)

[Cesari *et al.* 2017] G. Cesari, G. Schildbach, A. Carvalho et F. Borrelli. *Scenario model predictive control for lane change assistance and autonomous driving on highways.* IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 3, pages 23–35, 2017. (p. 85 and 102)

[Chebly *et al.* 2015] A. Chebly, G. Tagne, R. Talj et A. Charara. *Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 32 and 46)

[Chebly *et al.* 2017] A. Chebly, R. Talj et A. Charara. *Maneuver Planning for Autonomous Vehicles, with Clothoid Tentacles for Local Trajectory Planning.* In IEEE Int. Conf. on Intelligent Transportation (ITSC), 2017. (p. 85)

[Chen *et al.* 2013] J. Chen, P. Zhao, T. Mei et H. Liang. *Lane change path planning based on piecewise Bezier curve for autonomous vehicle.* In IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), 2013. (p. 31, 33 and 46)

[Chen *et al.* 2014] J. Chen, P. Zhao, H. Liang et T. Mei. *A Multiple Attribute-based Decision Making model for autonomous vehicle in urban environment.* In IEEE Intelligent Vehicles Symposium (IV), 2014. (p. 51)

[Chen *et al.* 2015a] C. Chen, A. Gaschler, M. Rickert et A. Knoll. *Task planning for highly automated driving.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 40 and 47)

[Chen *et al.* 2015b] C. Chen, A. Seff, A. Kornhauser et J. Xiao. *Deepdriving: Learning affordance for direct perception in autonomous driving.* In IEEE Int. Conf. on Computer Vision, 2015. (p. 38 and 47)

[Cherubini *et al.* 2012] A. Cherubini, F. Spindler et F. Chaumette. *A new tentacles-based technique for avoiding obstacles during visual navigation.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2012. (p. 31, 32, 46 and 85)

[Choi 2014] J. Choi. *Kinodynamic motion planning for autonomous vehicles.* International Journal of Advanced Robotic Systems, vol. 11, no. 6, page 90, 2014. (p. 25, 26, 32, 34, 46 and 47)

[CityMobil] CityMobil project website. http://www.citymobil-project.eu/ accessed 2018-11-03. (p. 3)

[Claussmann *et al.* 2015] L. Claussmann, A. Carvalho et G. Schildbach. *A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams.* In IEEE European Control Conference (ECC), 2015. (p. 25, 32, 35, 46 and 47)

[Connell & La 2017] D. Connell et H. M. La. *Dynamic path planning and replanning for mobile robots using RRT.* In IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC), 2017. (p. 29)

[Constantin *et al.* 2014] A. Constantin, J. Park et K. Iagnemma. *A margin-based approach to threat assessment for autonomous highway navigation.* In IEEE Intelligent Vehicles Symposium (IV), 2014. (p. 27, 35, 46 and 47)

[Da Lio *et al.* 2015] M. Da Lio, F. Biral, E. Bertolazzi, M. Galvani, P. Bosetti, D. Windridge, A. Saroldi et F. Tango. *Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems.* IEEE Trans. on Intelligent Transportation Systems, vol. 16, no. 1, pages 244–263, 2015. (p. 162)

[Daniel *et al.* 2013] J. Daniel, J.-P. Lauffenburger, S. Bernet et M. Basset. *Driving risk assessment with belief functions.* In IEEE Intelligent Vehicles Symposium (IV), 2013. (p. 40, 47, 51, 52 and 53)

[DARPA] DARPA Urban Challenge. http://archive.darpa.mil/grandchallenge/ accessed 2018-11-03. (p. 3)

[de Lima & Pereira 2013] D. A. de Lima et G. A. S. Pereira. *Navigation of an autonomous car using vector fields and the dynamic window approach.* Journal of Control, Automation and Electrical Systems, vol. 24, no. 1-2, pages 106–116, 2013. (p. 26, 30 and 46)

[Delling *et al.* 2009] D. Delling, P. Sanders, D. Schultes et D. Wagner. *Engineering route planning algorithms.* In Algorithmics of large and complex networks, pages 117–139. Springer, 2009. (p. 17)

[Dempster 1967] A. P. Dempster. *Upper and lower probabilities induced by a multivalued mapping.* The annals of mathematical statistics, pages 325–339, 1967. (p. 52 and 60)

[Derbel & Landry 2018] O. Derbel et R. J. Landry. *Belief and fuzzy theories for driving behavior assessment in case of accident scenarios.* Int. J. Automotive Technology, vol. 19, no. 1, pages 167–177, 2018. (p. 52)

[Dijkstra 1959] E. Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, vol. 1, no. 1, pages 269–271, 1959. (p. 28)

[Dolgov *et al.* 2008] D. Dolgov, S. Thrun, M. Montemerlo et J. Diebel. *Practical search techniques in path planning for autonomous driving.* Ann Arbor, vol. 1001, no. 48105, pages 18–80, 2008. (p. 28 and 46)

[Dubins 1957] L. E. Dubins. *On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents.* American Journal of mathematics, vol. 79, no. 3, pages 497–516, 1957. (p. 31)

[Eglese 1990] R. Eglese. *Simulated annealing: a tool for operational research.* European Journal of Operational Research, vol. 46, no. 3, pages 271–281, 1990. (p. 99)

[Elbanhawi *et al.* 2015] M. Elbanhawi, M. Simic et R. Jazar. *In the passenger seat: investigating ride comfort measures in autonomous cars.* IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 3, pages 4–17, 2015. (p. 162)

[Englund *et al.* 2016] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson et J. Didoff. *The Grand Cooperative Driving Challenge 2016: boosting the introduction of cooperative automated vehicles.* IEEE Wireless Communications, vol. 23, no. 4, pages 146–152, 2016. (p. 3)

[EPC] Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF accessed 2019-03-19. (p. 2)

[EUREKA] EUREKA Project E!45 PROMETHEUS. *Programme for a European traffic system with highest efficiency and unprecedented safety.* http://www.eurekanetwork.org/project/id/45 accessed 2018-11-03. (p. 2)

[Fiorini & Shiller 1998] P. Fiorini et Z. Shiller. *Motion planning in dynamic environments using velocity obstacles.* The International Journal of Robotics Research, vol. 17, no. 7, pages 760–772, 1998. (p. 26 and 85)

[Fox *et al.* 1997] D. Fox, W. Burgard et S. Thrun. *The dynamic window approach to collision avoidance.* IEEE Robotics & Automation Magazine, vol. 4, no. 1, pages 23–33, 1997. (p. 26)

[Fraichard & Asama 2004] T. Fraichard et H. Asama. *Inevitable collision states—A step towards safer robots?* Advanced Robotics, vol. 18, no. 10, pages 1001–1024, 2004. (p. 85)

[Furda & Vlacic 2010] A. Furda et L. Vlacic. *Multiple criteria-based real-time decision making by autonomous city vehicles.* IFAC Proceedings Volumes, vol. 43, no. 16, pages 97–102, 2010. (p. 51, 53, 54 and 74)

[Galceran *et al.* 2015] E. Galceran, R. M. Eustice et E. Olson. *Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 29, 30 and 46)

[Galceran *et al.* 2017] E. Galceran, A. G. Cunningham, R. M. Eustice et E. Olson. *Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment.* Autonomous Robots, vol. 41, no. 6, pages 1367–1382, 2017. (p. 36, 37 and 47)

[Gehrig & Stein 2007] S. K. Gehrig et F. J. Stein. *Collision avoidance for vehicle-following systems.* IEEE Trans. on Intelligent Transportation Systems, vol. 8, no. 2, pages 233–244, 2007. (p. 30 and 46)

[Gendreau & Potvin 2010] M. Gendreau et J.-Y. Potvin. Handbook of metaheuristics, volume 2. Springer, 2010. (p. 85)

[Geng *et al.* 2016] X. Geng, H. Liang, H. Xu, B. Yu et M. Zhu. *Human-driver speed profile modeling for autonomous vehicle's velocity strategy on curvy paths.* In IEEE Intelligent Vehicles Symposium (IV), 2016. (p. 38 and 47)

[Gerdes & Rossetter 2001] J. C. Gerdes et E. J. Rossetter. *A unified approach to driver assistance systems based on artificial potential fields.* Journal of Dynamic Systems, Measurement, and Control, vol. 123, no. 3, pages 431–438, 2001. (p. 29, 30 and 46)

[Ghumman *et al.* 2008] U. S. Ghumman, F. Kunwar, B. Benhabib*et al. Guidance-Based online motion planning for autonomous highway overtaking.* Int. J. on Smart Sensing and Intelligent Systems, vol. 1, no. 2, pages 549–571, 2008. (p. 39 and 47)

[Glaser *et al.* 2010] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer et L. Nouveliere. *Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction.* IEEE Trans. on Intelligent Transportation Systems, vol. 11, no. 3, pages 589–606, 2010. (p. 39 and 47)

[González *et al.* 2016] D. González, J. Pérez, V. Milanés et F. Nashashibi. *A Review of Motion Planning Techniques for Automated Vehicles.* IEEE Trans. on Intelligent Transportation Systems, vol. 17, no. 4, pages 1135–1145, 2016. (p. 7, 15 and 17)

[Goodall 2014] N. J. Goodall. *Ethical decision making during automated vehicle crashes.* Transportation Research Record, vol. 2424, no. 1, pages 58–65, 2014. (p. 163)

[Green 1925] H. Green. *Radio-Controlled Automobile.* Radio News, pages 592–656, Novembre 1925. (p. 2)

[Gruyer *et al.* 2017] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila et A. Rakotonirainy. *Perception, information processing and modeling: Critical stages for autonomous driving applications.* Annual Reviews in Control, 2017. (p. 20 and 42)

[Gu *et al.* 2013] T. Gu, J. Snider, J. Dolan et J.-w. Lee. *Focused trajectory planning for autonomous on-road driving.* In IEEE Intelligent Vehicles Symposium (IV), 2013. (p. 17, 27, 32, 33, 46 and 47)

[Gu *et al.* 2016a] T. Gu, J. M. Dolan et J.-W. Lee. *On-road trajectory planning for general autonomous driving with enhanced tunability.* In Intelligent Autonomous Systems, volume 13, pages 247–261. Springer, 2016. (p. 34 and 47)

[Gu *et al.* 2016b] T. Gu, J. M. Dolan et J.-W. Lee. *Runtime-bounded tunable motion planning for autonomous driving.* In IEEE Intelligent Vehicles Symposium (IV), 2016. (p. 26, 27, 28 and 46)

[Gündüz *et al.* 2017] G. Gündüz, Ç. Yaman, A. U. Peker et T. Acarman. *Driving pattern fusion using dempster-shafer theory for fuzzy driving risk level assessment.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 52 and 65)

[Hajek 1988] B. Hajek. *Cooling schedules for optimal annealing.* Mathematics of Operations Research, vol. 13, no. 2, pages 311–329, 1988. (p. 100)

[Hart *et al.* 1968] P. E. Hart, N. J. Nilsson et B. Raphael. *A formal basis for the heuristic determination of minimum cost paths.* IEEE Trans. on Systems Science and Cybernetics, vol. 4, no. 2, pages 100–107, 1968. (p. 28 and 33)

[Hasenjäger & Wersing 2017] M. Hasenjäger et H. Wersing. *Personalization in advanced driver assistance systems and autonomous vehicles: A review.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2017. (p. 5)

[HAVEit] HAVEit. *The future of driving. Deliverable D61.1 Final Report.* http://www.haveit-eu.org/LH2Uploads/ItemsContent/24/HAVEit_212154_D61.1_Final_Report_Published.pdf accessed 2018-11-03. (p. 3)

[He *et al.* 2018] X. He, Y. Liu, C. Lv, X. Ji et Y. Liu. *Emergency steering control of autonomous vehicle for collision avoidance and stabilisation.* Vehicle System Dynamics, pages 1–25, 2018. (p. 19)

[Hesse *et al.* 2010] T. Hesse, D. Hess et T. Sattel. *Motion planning for passenger vehicles-force field trajectory optimization for automated driving.* In Int. Conf. on Robotics and Applications, 2010. (p. 24, 28, 30 and 46)

[Hsu & Sun 2004] D. Hsu et Z. Sun. *Adaptively combining multiple sampling strategies for probabilistic roadmap planning.* In IEEE. Conf. on Robotics, Automation and Mechatronics, 2004. (p. 24)

[Huang *et al.* 1994] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. J. Russell et J. Weber. *Automatic symbolic traffic scene analysis using belief networks.* In AAAI, volume 94, pages 966–972, 1994. (p. 39 and 47)

[Hudecek & Eckstein 2014] J. Hudecek et L. Eckstein. *Improving and simplifying the generation of reference trajectories by usage of road-aligned coordinate systems.* In IEEE Intelligent Vehicles Symposium (IV), 2014. (p. 32 and 89)

[Huy *et al.* 2013] Q. Huy, S. Mita, H. T. N. Nejad et L. Han. *Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles.* IEICE Trans. on Information and Systems, vol. 96, no. 2, pages 314–328, 2013. (p. 33, 37, 46 and 47)

[Hwan Jeon *et al.* 2013] J. Hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras et K. Iagnemma. *Optimal motion planning with the half-car dynamical model for autonomous high-speed driving.* In IEEE American Control Conference (ACC), 2013. (p. 29 and 46)

[ICB 2015] Learn to drive smart: Your guide to driving safely. Insurance Corporation of British Columbia (ICBC), North Vancouver, B.C., Canada, 2015. (p. 76)

[Jafari *et al.* 2017] R. Jafari, S. Zeng, N. Moshchuk et B. Litkouhi. *Reactive path planning for emergency steering maneuvers on highway roads.* In IEEE American Control Conference (ACC), pages 2943–2949, 2017. (p. 19)

[Jang 1993] J.-S. Jang. *ANFIS: adaptive-network-based fuzzy inference system.* IEEE Trans. Systems, Man, and Cybernetics, vol. 23, no. 3, pages 665–685, 1993. (p. 58)

[Jiang *et al.* 2012] W. Jiang, D. Duanmu, X. Fan et Y. Deng. *A new method to determine basic probability assignment under fuzzy environment.* In IEEE Int. Conf. on Systems and Informatics (ICSAI), 2012. (p. 65)

[Johnson & Hauser 2013] J. Johnson et K. Hauser. *Optimal longitudinal control planning with moving obstacles.* In IEEE Intelligent Vehicles Symposium (IV), 2013. (p. 26 and 46)

[Jula *et al.* 2000] H. Jula, E. B. Kosmatopoulos et P. A. Ioannou. *Collision avoidance analysis for lane changing and merging.* IEEE Trans. on Vehicular Technology, vol. 49, no. 6, pages 2295–2308, 2000. (p. 96)

[Kang *et al.* 2015] Y. Kang, D. A. de Lima et A. C. Victorino. *Dynamic obstacles avoidance based on image-based dynamic window approach for human-vehicle interaction.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 26 and 46)

[Kant & Zucker 1986] K. Kant et S. W. Zucker. *Toward efficient trajectory planning: The path-velocity decomposition.* The international journal of robotics research, vol. 5, no. 3, pages 72–89, 1986. (p. 26)

[Katrakazas *et al.* 2015] C. Katrakazas, M. Quddus, W.-H. Chen et L. Deka. *Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions.* Transportation Research Part C: Emerging Technologies, vol. 60, pages 416–442, 2015. (p. 7)

[Kavraki *et al.* 1996] L. Kavraki, P. Svestka, J.-C. Latombe et M. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces.* IEEE Trans. on Robotics and Automation, vol. 12, no. 4, pages 566–580, 1996. (p. 23 and 24)

[Keller *et al.* 2014] M. Keller, F. Hoffmann, C. Hass, T. Bertram et A. Seewald. *Planning of optimal collision avoidance trajectories with timed elastic bands.* IFAC Proceedings Volumes, vol. 47, no. 3, pages 9822–9827, 2014. (p. 30, 33, 46 and 47)

[Khatib & Le Maitre 1978] O. Khatib et J. Le Maitre. *Dynamic control of manipulators operation in a complex environment.* In CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators, 1978. (p. 29 and 41)

[Khatib 1986] O. Khatib. *Real-time obstacle avoidance for manipulators and mobile robots.* In Autonomous Robot Vehicles, pages 396–404. Springer, 1986. (p. 29 and 30)

[Kirkpatrick *et al.* 1983] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi. *Optimization by simulated annealing.* Science, vol. 220, no. 4598, pages 671–680, 1983. (p. 97)

[Ko & Simmons 1998] N. Y. Ko et R. G. Simmons. *The lane-curvature method for local obstacle avoidance.* In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, volume 3, pages 1615–1621, 1998. (p. 27)

[Kuan *et al.* 1984] D. T. Kuan, R. A. Brooks, J. C. Zamiska et M. Das. *Automatic path planning for a mobile robot using a mixed representation of free space.* In IEEE Computer Society Conference on Artificial Intelligence Applications, 1984. (p. 26)

[Kuan *et al.* 1985] D. Kuan, J. Zamiska et R. Brooks. *Natural decomposition of free space for path planning.* In IEEE Int. Conf. on Robotics and Automation, 1985. (p. 24 and 26)

[Latombe 1991] J.-C. Latombe. Robot motion planning. Springer Science & Business Media, 1991. (p. 7)

[Lattarulo *et al.* 2017] R. Lattarulo, M. Marcano et J. Pérez. *Overtaking maneuver for automated driving using virtual environments.* In Int. Conf. on Computer Aided Systems Theory, pages 446–453. Springer, 2017. (p. 85)

[Laumond *et al.* 1998] J.-P. Laumond, S. Sekhavat et F. Lamiraux. *Guidelines in nonholonomic motion planning for mobile robots.* In Robot motion planning and control, pages 1–53. Springer, 1998. (p. 7)

[LaValle 1998] S. M. LaValle. *Rapidly-exploring random trees: A new tool for path planning.* 1998. (p. 29)

[LaValle 2006] S. M. LaValle. Planning algorithms. Cambridge university press, 2006. (p. 7 and 27)

[Lee *et al.* 2014] U. Lee, S. Yoon, H. Shim, P. Vasseur et C. Demonceaux. *Local path planning in a complex environment for self-driving car.* In IEEE Int. Conf. on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014. (p. 25, 26 and 46)

[Lefèvre *et al.* 2014] S. Lefèvre, D. Vasquez et C. Laugier. *A survey on motion prediction and risk assessment for intelligent vehicles.* Robomech Journal, vol. 1, no. 1, pages 1–14, 2014. (p. 17, 39 and 51)

[Lefevre *et al.* 2016] S. Lefevre, A. Carvalho et F. Borrelli. *A learning-based framework for velocity control in autonomous driving.* IEEE Trans. on Automation Science and Engineering, vol. 13, no. 1, pages 32–42, 2016. (p. 33, 38 and 47)

[Li *et al.* 2014] X. Li, Z. Sun, Q. Zhu et D. Liu. *A unified approach to local trajectory planning and control for autonomous driving along a reference path.* In IEEE Int. Conf. on Mechatronics and Automation (ICMA), 2014. (p. 24, 33, 46 and 47)

[Li *et al.* 2017] N. Li, H. Chen, I. Kolmanovsky et A. Girard. *An Explicit Decision Tree Approach for Automated Driving.* In ASME Dynamic Systems and Control Conference, 2017. (p. 35 and 47)

[Li *et al.* 2018] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky et A. R. Girard. *Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems.* IEEE Trans. on Control Systems Technology, vol. 26, no. 5, pages 1782–1797, 2018. (p. 36, 37, 38, 40 and 47)

[Lima *et al.* 2015a] P. F. Lima, M. Trincavelli, J. Mårtensson et B. Wahlberg. *Clothoid-based model predictive control for autonomous driving.* In IEEE European Control Conference (ECC), 2015. (p. 32, 33, 46 and 47)

[Lima *et al.* 2015b] P. F. Lima, M. Trincavelli, J. Mårtensson et B. Wahlberg. *Clothoid-based speed profiler and control for autonomous driving.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2015. (p. 31, 32 and 46)

[Liu *et al.* 2012] Y. Liu, W.-L. Huang, T. Sun et F. Zhu. *AGV decision making subsystem based on modified Dempster-Shafer evidence theory and fuzzy logic.* In IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), 2012. (p. 52, 53 and 66)

[Liu *et al.* 2017a] C. Liu, S. Lee, S. Varnhagen et H. E. Tseng. *Path planning for autonomous vehicles using model predictive control.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 24, 30, 33, 46 and 47)

[Liu *et al.* 2017b] C. Liu, W. Zhan et M. Tomizuka. *Speed profile planning in dynamic environments via temporal optimization.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 85)

[Lozano-Pérez & Wesley 1979] T. Lozano-Pérez et M. A. Wesley. *An algorithm for planning collision-free paths among polyhedral obstacles.* Communications of the ACM, vol. 22, no. 10, pages 560–570, 1979. (p. 26)

[Ma *et al.* 2015] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma et N. Zheng. *Efficient sampling-based motion planning for on-road autonomous driving.* IEEE Trans. on Intelligent Transportation Systems, vol. 16, no. 4, pages 1961–1976, 2015. (p. 29 and 46)

[McKnight & Adams 1970a] A. J. McKnight et B. B. Adams. *Driver Education Task Analysis. Volume II: Task Analysis Methods.* 1970. (p. 40 and 47)

[McKnight & Adams 1970b] A. McKnight et B. Adams. *Driver education task analysis. Volume I: Task descriptions.* 1970. (p. 40 and 47)

[McKnight & Hundt 1971a] A. J. McKnight et A. G. Hundt. *Driver Education Task Analysis. Volume IV: The Development of Instructional Objectives.* 1971. (p. 40 and 47)

[McKnight & Hundt 1971b] A. McKnight et A. Hundt. *Driver education task analysis. Volume III: Instructional Objectives.* 1971. (p. 40 and 47)

[McNaughton *et al.* 2011] M. McNaughton, C. Urmson, J. M. Dolan et J.-W. Lee. *Motion planning for autonomous driving with a conformal spatiotemporal lattice.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2011. (p. 27, 32 and 46)

[Merat *et al.* 2014] N. Merat, A. H. Jamson, F. C. Lai, M. Daly et O. M. Carsten. *Transition to manual: Driver behaviour when resuming control from a highly automated vehicle.* Transportation research part F: traffic psychology and behaviour, vol. 27, pages 274–282, 2014. (p. 162)

[Michon 1985] J. A. Michon. *A critical view of driver behavior models: what do we know, what should we do?* In Human behavior and traffic safety, pages 485–524. Springer, 1985. (p. 16 and 39)

[Milanés *et al.* 2011] V. Milanés, E. Onieva, J. P. Rastelli, J. Godoy et J. Villagrá. *An approach to driverless vehicles in highways.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2011. (p. 38 and 47)

[Miller *et al.* 2018] C. Miller, C. Pek et M. Althoff. *Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles.* In IEEE Intelligent Vehicles Symposium (IV), 2018. (p. 33 and 47)

[Mitsch *et al.* 2013] S. Mitsch, K. Ghorbal et A. Platzer. *On provably safe obstacle avoidance for autonomous robotic ground vehicles.* In Robotics: Science and Systems (RSS), 2013. (p. 26 and 46)

[Moras *et al.* 2011] J. Moras, V. Cherfaoui et P. Bonnifait. *Credibilist occupancy grids for vehicle perception in dynamic environments.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2011. (p. 25 and 46)

[Morignot & Nashashibi 2013] P. Morignot et F. Nashashibi. *An ontology-based approach to relax traffic regulation for autonomous vehicle assistance.* IASTED Int. Conf. on Artificial Intelligence and Applications, 2013. (p. 163)

[Mouhagir *et al.* 2016] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun et F. Guillemard. *Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2016. (p. 25, 32, 36, 46 and 47)

[Naranjo *et al.* 2008] J. E. Naranjo, C. Gonzalez, R. Garcia et T. De Pedro. *Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver.* IEEE Trans. on Intelligent Transportation Systems, vol. 9, no. 3, pages 438–450, 2008. (p. 38, 40 and 47)

[Naumann & Stiller 2017] M. Naumann et C. Stiller. *Towards cooperative motion planning for automated vehicles in mixed traffic.* IEEE Int. Conf. on Intelligent Robots and Systems (IROS) Workshops, 2017. (p. 39 and 47)

[Ngai & Yung 2011] D. C. K. Ngai et N. H. C. Yung. *A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers.* IEEE Trans. on Intelligent Transportation Systems, vol. 12, no. 2, pages 509–522, 2011. (p. 38 and 47)

[NHA] *"No Hands Across America" Project.* http://www.cs.cmu.edu/~tjochem/nhaa/nhaa_home_page.html accessed 2018-11-03. (p. 3)

[Nilsson *et al.* 2015] J. Nilsson, J. Fredriksson et E. Coelingh. *Rule-based highway maneuver intention recognition.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2015. (p. 35 and 47)

[Nilsson *et al.* 2016] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh et J. Fredriksson. *If, when, and how to perform lane change maneuvers on highways.* IEEE Intelligent Transportation Systems Magazine, vol. 8, no. 4, pages 68–78, 2016. (p. 35 and 47)

[Nilsson *et al.* 2017] J. Nilsson, J. Fredriksson et E. Coelingh. *Trajectory planning with miscellaneous safety critical zones.* IFAC-PapersOnLine, vol. 50, no. 1, pages 9083–9088, 2017. (p. 24, 33 and 47)

[Noh & An 2018] S. Noh et K. An. *Decision-Making Framework for Automated Driving in Highway Environments.* IEEE Trans. Intelligent Transportation Systems, vol. 19, no. 1, pages 58–71, 2018. (p. 51 and 64)

[Ok *et al.* 2013] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert et M. Stilman. *Path planning with uncertainty: Voronoi uncertainty fields.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2013. (p. 26)

[Onieva *et al.* 2011] E. Onieva, J. E. Naranjo, V. Milanés, J. Alonso, R. García et J. Pérez. *Automatic lateral control for unmanned vehicles via genetic algorithms.* Applied Soft Computing, vol. 11, no. 1, pages 1303–1309, 2011. (p. 37 and 47)

[O' Brien *et al.* 2017] M. O' Brien, K. Neubauer, J. Van Brummelen et H. Najjaran. *Analysis of Driving Data for Autonomous Vehicle Applications.* In IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC), 2017. (p. 76)

[Paden *et al.* 2016] B. Paden, M. Čáp, S. Z. Yong, D. Yershov et E. Frazzoli. *A survey of motion planning and control techniques for self-driving urban vehicles.* IEEE Trans. on Intelligent Vehicles, vol. 1, no. 1, pages 33–55, 2016. (p. 7 and 41)

[Payre *et al.* 2014] W. Payre, J. Cestac et P. Delhomme. *Intention to use a fully automated car: Attitudes and a priori acceptability.* Transportation Research Part F: Traffic Psychology and Behaviour, vol. 27, pages 252–263, 2014. (p. 7)

[PC6 1960] *Reporter Rides Driverless Car.* The Press-Courier, Juin 7 1960. (p. 2)

[Pek *et al.* 2017a] C. Pek, M. Koschi, M. Werling et M. Althoff. *Enhancing motion safety by identifying safety-critical passageways.* In IEEE Annual Conf. on Decision and Control (CDC), 2017. (p. 161)

[Pek *et al.* 2017b] C. Pek, P. Zahn et M. Althoff. *Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 161)

[Perez-Lozano 1983] T. Perez-Lozano. *Spatial planning: A configuration space approach.* IEEE Trans. on Computers, vol. 32, 1983. (p. 21)

[Pivtoraiko *et al.* 2009] M. Pivtoraiko, R. A. Knepper et A. Kelly. *Differentially constrained mobile robot motion planning in state lattices.* Journal of Field Robotics, vol. 26, no. 3, pages 308–333, 2009. (p. 27)

[Plessen *et al.* 2017] M. G. Plessen, P. F. Lima, J. Mårtensson, A. Bemporad et B. Wahlberg. *Trajectory planning under vehicle dimension constraints using sequential linear programming.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2017. (p. 33, 47 and 85)

[Polack *et al.* 2017] P. Polack, F. Altché, B. d'Andréa Novel et A. de La Fortelle. *The Kinematic Bicycle Model: a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 89)

[Polack *et al.* 2018] P. Polack, F. Altché, B. D'Andrea-Novel et A. de La Fortelle. *Guaranteeing Consistency in a Motion Planning and Control Architecture Using a Kinematic Bicycle Model.* In IEEE American Control Conference (ACC), 2018. (p. 20)

[Prokhorov 2009] D. Prokhorov. *Risk estimator for control in intelligent transportation system.* In IEEE Control Applications (CCA) & Intelligent Control (ISIC), 2009. (p. 40, 47, 51, 53 and 74)

[Qian *et al.* 2016] X. Qian, F. Altché, P. Bender, C. Stiller et A. de La Fortelle. *Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2016. (p. 33, 47 and 51)

[Qu *et al.* 2015] P. Qu, J. Xue, L. Ma et C. Ma. *A constrained vfh algorithm for motion planning of autonomous vehicles.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 25 and 46)

[Quilbeuf *et al.* 2018] J. Quilbeuf, M. Barbier, L. Rummelhard, C. Laugier, A. Legay, B. Baudouin, T. Genevois, J. Ibañez-Guzmán et O. Simonin. *Statistical Model Checking Applied on Perception and Decision-making Systems for Autonomous Driving.* In IEEE Int. Conf. on Intelligent Robots and Systems (IROS) Workshops, 2018. (p. 163)

[Quinlan & Khatib 1993] S. Quinlan et O. Khatib. *Elastic bands: Connecting path planning and control.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 1993. (p. 30)

[Rajamani *et al.* 2000] R. Rajamani, S. Choi, B. Law, J. K. Hedrick, R. Prohaska et P. Kretz. *Design and experimental implementation of longitudinal control for a platoon of automated vehicles.* J. of Dynamic Systems, Measurement, and Control, vol. 122, no. 3, pages 470–476, 2000. (p. 3)

[Rasekhipour *et al.* 2017] Y. Rasekhipour, A. Khajepour, S.-K. Chen et B. Litkouhi. *A potential field-based model predictive path-planning controller for autonomous road vehicles.* IEEE Trans. on Intelligent Transportation Systems, vol. 18, no. 5, pages 1255–1267, 2017. (p. 30 and 46)

[Rehder *et al.* 2017] E. Rehder, J. Quehl et C. Stiller. *Driving Like a Human: Imitation Learning for Path Planning using Convolutional Neural Networks.* In IEEE Int. Conf. on Intelligent Robots and Systems (IROS) Workshops, 2017. (p. 38 and 47)

[Reichardt & Shick 1994] D. Reichardt et J. Shick. *Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway.* In IEEE Intelligent Vehicles Symposium (IV), 1994. (p. 29 and 46)

[Reif & Dietsche 2010] K. Reif et K.-H. Dietsche. Kraftfahrtechnisches taschenbuch. Springer-Verlag, 2010. (p. 89, 90, 106, 127 and 129)

[Resende *et al.* 2013] P. Resende, E. Pollard, H. Li et F. Nashashibi. *ABV – A Low Speed Automation Project to Study the Technical Feasibility of Fully Automated Driving.* In Workshop PAMM, 2013. (p. 3)

[Rezaei *et al.* 2003] S. Rezaei, J. Guivant et E. M. Nebot. *Car-like robot path following in large unstructured environments.* In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2003. (p. 29 and 46)

[Riaz *et al.* 2015] F. Riaz, M. A. Niazi, M. Sajid, S. Amin, N. I. Ratyal et F. Butt. *An efficient collision avoidance scheme for autonomous vehicles using genetic algorithm.* J. Appl. Environ. Biol. Sci, vol. 5, no. 8, pages 70–76, 2015. (p. 37 and 47)

[Rodrigues *et al.* 2016] M. Rodrigues, A. McGordon, G. Gest et J. Marco. *Adaptive tactical behaviour planner for autonomous ground vehicle.* In IEEE Int. Conf. on Control, 2016. (p. 15, 17, 20, 36 and 47)

[Russell *et al.* 1995] S. Russell, P. Norvig et A. Intelligence. *A modern approach.* Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 1995. (p. 34)

[SAE] SAE International. https://www.sae.org/standards/content/j3016_201401/ accessed 2018-11-03. (p. 4 and 5)

[Salvucci 2006] D. D. Salvucci. *Modeling driver behavior in a cognitive architecture.* Human factors, vol. 48, no. 2, pages 362–380, 2006. (p. 39)

[SARTRE] SARTRE project website. https://www.sp.se/sv/index/research/dependable_systems/Documents/The%20SARTRE%20project.pdf accessed 2018-11-03. (p. 3)

[Schubert *et al.* 2008] R. Schubert, U. Scheunert et G. Wanielik. *Planning feasible vehicle manoeuvres on highways.* IET Intelligent Transport Systems, vol. 2, no. 3, pages 211–218, 2008. (p. 27, 28, 32 and 46)

[Schürmann *et al.* 2017] B. Schürmann, D. Hess, J. Eilbrecht, O. Stursberg, F. Köster et M. Althoff. *Ensuring drivability of planned motions using formal methods.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2017. (p. 20)

[Sentz *et al.* 2002] K. Sentz, S. Ferson*et al.* Combination of evidence in dempster-shafer theory, volume 4015. Citeseer, 2002. (p. 60 and 66)

[Shafer 1976] G. Shafer. A mathematical theory of evidence, volume 42. Princeton university press, 1976. (p. 52 and 59)

[Sheckells *et al.* 2017] M. Sheckells, T. M. Caldwell et M. Kobilarov. *Fast approximate path coordinate motion primitives for autonomous driving.* In IEEE Annual Conf. on Decision and Control (CDC), 2017. (p. 32 and 46)

[Song *et al.* 2013] X. Song, H. Cao et J. Huang. *Vehicle path planning in various driving situations based on the elastic band theory for highway collision avoidance.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 227, no. 12, pages 1706–1722, 2013. (p. 25, 30 and 46)

[Söntges & Althoff 2018] S. Söntges et M. Althoff. *Computing the drivable area of autonomous road vehicles in dynamic road scenes.* IEEE Trans. on Intelligent Transportation Systems, vol. 19, no. 6, pages 1855–1866, 2018. (p. 85 and 86)

[Stentz 1994] A. Stentz. *Optimal and efficient path planning for partially-known environments.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 1994. (p. 29)

[Sugihara 1993] K. Sugihara. *Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams.* CVGIP: Graphical Models and Image Processing, vol. 55, no. 6, pages 522–531, 1993. (p. 26)

[Tamke *et al.* 2011] A. Tamke, T. Dang et G. Breuel. *A flexible method for criticality assessment in driver assistance systems.* In IEEE Intelligent Vehicles Symposium (IV), 2011. (p. 39 and 47)

[Taylor 1964] D. Taylor. *Drivers' galvanic skin response and the risk of accident.* Ergonomics, vol. 7, no. 4, pages 439–451, 1964. (p. 39 and 47)

[Tehrani *et al.* 2015] H. Tehrani, Q. H. Do, M. Egawa, K. Muto, K. Yoneda et S. Mita. *General behavior and motion model for automated lane change.* In IEEE Intelligent Vehicles Symposium (IV), 2015. (p. 25, 32, 35, 46, 47 and 163)

[Thornton *et al.* 2017] S. M. Thornton, S. Pan, S. M. Erlien et J. C. Gerdes. *Incorporating ethical considerations into automated vehicle control.* IEEE Trans. on Intelligent Transportation Systems, vol. 18, no. 6, pages 1429–1439, 2017. (p. 163)

[Trepagnier *et al.* 2011] P. G. Trepagnier, J. E. Nagel, P. M. Kinney, M. T. Dooner, B. M. Wilson, C. R. Schneider Jr et K. B. Goeller. *Navigation and control system for autonomous vehicles*, Novembre 1 2011. US Patent 8,050,863. (p. 26 and 46)

[Tsugawa *et al.* 1979] S. Tsugawa, T. Yatabe, T. Hirose et S. Matsumoto. *An automobile with artificial intelligence.* In Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2, pages 893–895. Morgan Kaufmann Publishers Inc., 1979. (p. 2)

[Ulbrich & Maurer 2015] S. Ulbrich et M. Maurer. *Towards tactical lane change behavior planning for automated vehicles.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2015. (p. 36, 47 and 161)

[Ulmer 1994] B. Ulmer. *Vita ii-active collision avoidance in real traffic.* In IEEE Intelligent Vehicles Symposium (IV), 1994. (p. 2)

[UNECE] UNECE press. https://www.unece.org/info/media/presscurrent-press-h/transport/2016/unece-paves-the-way-for-automated-driving-by-updating-un-international-convention/doc.html accessed 2018-11-03. (p. 4)

[United Nations] United Nations. *Convention on Road Traffic (Vienna).* https://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf accessed 2018-11-03. (p. 4 and 91)

[Vallon *et al.* 2017] C. Vallon, Z. Ercan, A. Carvalho et F. Borrelli. *A machine learning approach for personalized autonomous lane change initiation and control.* In IEEE Intelligent Vehicles Symposium (IV), 2017. (p. 37 and 47)

[Vanholme *et al.* 2010] B. Vanholme, D. Gruyer, S. Glaser et S. Mammar. *Fast prototyping of a highly autonomous cooperative driving system for public roads.* In IEEE Intelligent Vehicles Symposium (IV), 2010. (p. 19)

[Vanholme *et al.* 2013] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser et S. Mammar. *Highly automated driving on highways based on legal safety.* IEEE Trans. on Intelligent Transportation Systems, vol. 14, no. 1, pages 333–347, 2013. (p. 31, 32, 35, 46 and 47)

[Verberne *et al.* 2012] F. M. Verberne, J. Ham et C. J. Midden. *Trust in smart systems: Sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars.* Human factors, vol. 54, no. 5, pages 799–810, 2012. (p. 162)

[Villagra *et al.* 2012] J. Villagra, V. Milanés, J. P. Rastelli, J. Godoy et E. Onieva. *Path and speed planning for smooth autonomous navigation.* In IEEE Intelligent Vehicles Symposium (IV), 2012. (p. 24, 26, 27, 28, 29, 31, 32 and 46)

[Von Hundelshausen *et al.* 2008] F. Von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller et H.-J. Wuensche. *Driving with tentacles: Integral structures for sensing and motion.* Journal of Field Robotics, vol. 25, no. 9, pages 640–673, 2008. (p. 31 and 85)

[Wagner *et al.* 2018] S. Wagner, K. Groh, T. Kühbeck, M. Dörfel et A. Knol. *Using Time-to-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving.* In IEEE Intelligent Vehicles Symposium (IV), 2018. (p. 39 and 47)

[Wang & Ayalew 2016] Q. Wang et B. Ayalew. *Obstacle filtering algorithm for control of an autonomous road vehicle in public highway traffic.* In ASME Dynamic Systems and Control Conference, 2016. (p. 25, 33, 39 and 47)

[Wang *et al.* 2015] Q. Wang, T. Weiskircher et B. Ayalew. *Hierarchical hybrid predictive control of an autonomous road vehicle.* In ASME Dynamic Systems and Control Conference, 2015. (p. 35, 36 and 47)

[Wang *et al.* 2017] W. Wang, C. Liu et D. Zhao. *How much data are enough? A statistical approach with case study on longitudinal driving behavior.* IEEE Trans. on Intelligent Vehicles, vol. 2, no. 2, pages 85–98, 2017. (p. 161)

[Wang *et al.* 2018a] P. Wang, C.-Y. Chan et A. de La Fortelle. *A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers.* IEEE Intelligent Vehicles Symposium (IV), 2018. (p. 38 and 47)

[Wang *et al.* 2018b] Q. Wang, B. Ayalew et T. Weiskircher. *Predictive maneuver planning for an autonomous vehicle in public highway traffic.* IEEE Trans. on Intelligent Transportation Systems, no. 99, pages 1–13, 2018. (p. 85)

[Wang *et al.* 2018c] Z. Wang, G. Wu et M. J. Barth. *A Review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), pages 2884–2891, 2018. (p. 6)

[Wang *et al.* 2019] H. Wang, Y. Huang, A. Khajepour, Y. Rasekhipour, Y. Zhang et D. Cao. *Crash Mitigation in Motion Planning for Autonomous Vehicles.* IEEE Trans. on Intelligent Transportation Systems, 2019. (p. 19)

[Wardziński 2006] A. Wardziński. *The role of situation awareness in assuring safety of autonomous vehicles.* Computer Safety, Reliability, and Security, pages 205–218, 2006. (p. 51, 52 and 64)

[Wilde 1982] G. J. Wilde. *The theory of risk homeostasis: implications for safety and health.* Risk analysis, vol. 2, no. 4, pages 209–225, 1982. (p. 39 and 47)

[Wolf & Burdick 2008] M. T. Wolf et J. W. Burdick. *Artificial potential functions for highway driving with collision avoidance.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2008. (p. 29, 30 and 46)

[Wolpert & Macready 1997] D. H. Wolpert et W. G. Macready. *No free lunch theorems for optimization.* IEEE Trans. on Evolutionary Computation, vol. 1, no. 1, pages 67–82, 1997. (p. 86)

[Xu *et al.* 2012] W. Xu, J. Wei, J. M. Dolan, H. Zhao et H. Zha. *A real-time motion planner with trajectory optimization for autonomous vehicles.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2012. (p. 27, 32, 33, 46, 47 and 85)

[Xu *et al.* 2014] W. Xu, J. Pan, J. Wei et J. M. Dolan. *Motion planning under uncertainty for on-road autonomous driving.* In IEEE Int. Conf. on Robotics and Automation (ICRA), 2014. (p. 161)

[Yager & Fileu 1993] R. R. Yager et D. P. Fileu. *Learning of fuzzy rules by mountain clustering.* In Applications of Fuzzy Logic Technology, volume 2061, pages 246–255. International Society for Optics and Photonics, 1993. (p. 58)

[Yager & Filev 1994] R. R. Yager et D. P. Filev. *Essentials of fuzzy modeling and control.* New York, vol. 388, 1994. (p. 51, 56 and 58)

[Yang *et al.* 2018] L. Yang, X. Liang, T. Wang et E. Xing. *Real-to-Virtual Domain Unification for End-to-End Autonomous Driving.* In Proceedings of the European Conference on Computer Vision (ECCV), 2018. (p. 38 and 47)

[Yao *et al.* 2012] W. Yao, H. Zhao, F. Davoine et H. Zha. *Learning lane change trajectories from on-road driving data.* In IEEE Intelligent Vehicles Symposium (IV), 2012. (p. 27 and 46)

[Yen 1990] J. Yen. *Generalizing the Dempster-Schafer theory to fuzzy sets.* IEEE Trans. Systems, Man, and Cybernetics, vol. 20, no. 3, pages 559–570, 1990. (p. 68)

[Yu *et al.* 2016] C. Yu, V. Cherfaoui et P. Bonnifait. *Semantic evidential lane grids with prior maps for autonomous navigation.* In IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2016. (p. 25, 46 and 161)

[Zadeh 1965] L. A. Zadeh. *Information and control.* Fuzzy sets, vol. 8, no. 3, pages 338–353, 1965. (p. 56)

[Zhou *et al.* 2017] S. Zhou, Y. Wang, M. Zheng et M. Tomizuka. *A hierarchical planning and control framework for structured highway driving.* IFAC-PapersOnLine, vol. 50, no. 1, pages 9101–9107, 2017. (p. 36 and 47)

[Ziegler & Stiller 2009] J. Ziegler et C. Stiller. *Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios.* In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2009. (p. 27, 32 and 46)

[Ziegler *et al.* 2014] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller*et al.* *Making Bertha Drive – An Autonomous Journey on a Historic Route.* IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 2, pages 8–20, 2014. (p. 3, 15, 17, 26, 33, 35, 46 and 47)

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Architecture unifiée de prise de décision et génération de trajectoires pour un véhicule autonome sur autoroute

Mots clés : Véhicule autonome, Planification de mouvement, Prise de décision multi-critères, Génération de trajectoires, Systèmes de transports intelligents

Résumé : Ce travail de thèse s'inscrit dans le développement d'un véhicule autonome en milieu autoroutier. Plus précisément, il s'agit de proposer une architecture unifiée de génération de trajectoires avec une prise de décision prenant en compte les limitations de l'environnement et des informations disponibles actuellement sur un véhicule automatisé.
La méthode propose d'une part de générer des trajectoires sous forme de sigmoïde dans une représentation spatiotemporelle continue de l'espace de navigation, préalablement réduit par la modélisation d'intervalles sans collision en condition nominale de conduite. Les paramètres de la sigmoïde sont ensuite optimisés par une stratégie de recuit simulé utilisant l'algorithme de prise de décision comme fonction d'évaluation de la trajectoire générée. De cette manière, les problèmes de discrétisation et de découplage position/vitesse sont évités. D'autre part, l'agrégation des théories de logique floue et des croyances permet une prise de décision sur des critères hétérogènes et des données incertaines. Le formalisme présenté offre la possibilité d'adapter le comportement du véhicule aux passagers, notamment selon leur perception du risque et leur souhait d'une conduite souple ou sportive.
L'approche développée a finalement été évaluée et validée en environnement de simulation puis sur un véhicule de test. La brique de planification a alors été intégrée à l'architecture existante du véhicule, en aval des briques de localisation et de perception des obstacles et en amont de la brique de contrôle.

Title : Motion Planning for Autonomous Highway Driving: A Unified Architecture for Decision-Maker and Trajectory Generator

Keywords : Self-driving cars, Motion planning, Decision making, Trajectory generation, Intelligent Transportation Systems

Abstract : This thesis work is part of the development of a self-driving car in highway environments. More precisely, it aims to propose a unified architecture of trajectory planner and decision-maker taking into account the limitations of the environment and the available data within the current development of sensors technologies (distance limitations, uncertainties).
On the one hand, the method generates sigmoid trajectories in a continuous spatiotemporal representation of the evolution space, which is reduced beforehand by modeling collision-free intervals in nominal conditions of driving. The sigmoid parameters are subsequently optimized with a simulated annealing approach that uses the decision-maker algorithm as the evaluation function for the generated trajectory. It thus makes it possible to elude both the discretization and position/speed decoupling problems. On the other hand, the aggregation of fuzzy logic and belief theory allows decision making on heterogeneous criteria and uncertain data. The proposed framework also handles personalization of the vehicle's behavior, depending on the passengers' risk perception and an aggressive or conservative driving style.
The presented approach was finally evaluated and validated in a simulation environment, and then in a test vehicle. The planning block was integrated into the existing vehicle's architecture, interfaced with the localization, obstacles' perception and control blocks.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France