UNIVERSITÉ CLERMONT AUVERGNE

ÉCOLE DOCTORALE

SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

THÈSE

Présentée par

DIMIA IBERRAKEN

Master Robotique

de l'Université de Technologie de Compiègne

pour obtenir le grade de

Docteur d'Université

Spécialité : **Electronique et Systèmes**

Safe Trajectories and Sequential Bayesian Decision-Making Architecture for Reliable Autonomous Vehicle Navigation

Soutenue publiquement le 20 October 2020 devant le Jury composé de :

THIERRY FRAICHARD	Rapporteur	Chargé de recherche - HDR, INRIA Grenoble Rhône-Alpes
PHILIPPE MARTINET	Rapporteur	Directeur de recherche, INRIA Sophia-Antipolis Méditerranée
MATTHIAS ALTHOFF	Examinateur	Professeur, Technische Universität München
PHILIPPE BONNIFAIT	Examinateur	Professeur, Université de Technologie de Compiègne
OUIDDAD LABBANI-IGBIDA	Examinatrice	Professeur, Université de Limoges
YOUCEF MEZOUAR	Examinateur	Professeur, SIGMA Clermont
LOUNIS ADOUANE	Directeur de thèse	Professeur, Université de Technologie de Compiègne
DIEUMET DENIS	Invité	Chef de Projet, Sherpa Engineering

Abstract

Recent advances in Autonomous Vehicles (AV) driving raised up all the importance to ensure the complete reliability of AV maneuvers even in highly dynamic and uncertain environments/situations. This objective becomes even more challenging due to the uniqueness of every traffic situation/condition. To cope with all these very constrained and complex configurations, AVs must have appropriate control architecture with reliable and real-time Risk Assessment and Management Strategies (RAMS). These targeted RAMS must lead to reduce drastically the navigation risks (theoretically, lower than any human-like driving behavior), with a systemic way. Consequently, the aim is also to reduce the need for too extensive testing (which could take several months and years for each produced RAMS without at the end having absolute prove). Hence the goal in this Ph.D. thesis is to have a provable methodology for AV RAMS. This dissertation addresses the full pipeline from risk assessment, path planning to decision-making and control of autonomous vehicles. In the first place, an overall Probabilistic Multi-Controller Architecture (P-MCA) is designed for safe autonomous driving under uncertainties. The P-MCA is composed of several interconnected modules that are responsible for: assessing the collision risk with all observed vehicles while considering their trajectories' predictions; planning the different driving maneuvers; making the decision on the most suitable actions to achieve; control the vehicle movement; aborting safely the engaged maneuver if necessary (due for instance to a sudden change in the environment); and as last resort planning evasive actions if there is no other choice. The proposed risk assessment is based on a dual-safety stage strategy. The first stage analyzes the actual driving situation and predicts potential collisions. This is performed while taking into consideration several dynamic constraints and traffic conditions that are known at the time of planning. The second stage is applied in real-time, during the maneuver achievement, where a safety verification mechanism is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver. The decision-making strategy is based on a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) and corresponds to an important module of the P-MCA. This module is designed to manage several road maneuvers under uncertainties. It utilizes the defined safety stages assessment to propose discrete actions that allow to: derive appropriate maneuvers in a given traffic situation and provide a safety retrospection that updates in real-time the ego-vehicle movements according to the environment dynamic, in order to face any sudden hazardous and risky situation. In the latter case, it is proposed to compute the corresponding low-level control based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) that allows the ego-vehicle to pursue the advised collision-free evasive trajectory to avert an accident and to guarantee safety at any time. The reliability and the flexibility of the overall proposed P-MCA and its elementary components have been intensively validated, first in simulated traffic conditions, with various driving scenarios, and secondly, in real-time with the autonomous vehicles available at Institut Pascal.

Keywords: Autonomous driving, Multi-controller architectures, Risk assessment and management, Decision-making under uncertainty, Motion-planning, Safety verification.

Résumé

Les dernières avancées en matière de conduite de véhicules autonomes (VAs) ont fait apparaître toute l'importance de garantir la fiabilité complète des manœuvres que doivent effectuer les VAs, y compris dans des environnements/situations très dynamiques et incertains. Cet objectif devient encore plus ardu en raison du caractère unique de chaque situation/condition de circulation. Pour faire face à toutes ces configurations très contraignantes et complexes, les VAs doivent disposer d'une architecture de contrôle appropriée avec des Stratégies d'Evaluation et de Gestion des Risques (SEGR) fonctionnant en temps-réel et d'une manière fiable. Ces SEGR ciblées doivent conduire à une réduction drastique des risques de conduite. Théoriquement et de maniéré systémique, ces SEGR doivent aboutir à un risque de conduite inférieur à tout comportement de conduite humaine. En conséquent, il est également question de réduire la nécessité d'effectuer des tests très poussés, qui peuvent prendre plusieurs mois/années pour au final ne pas avoir de preuves formelles de la viabilité et de la sûreté complète du système. Ainsi, les travaux présentés dans cette thèse de doctorat ont pour but d'avoir une méthodologie prouvable pour les SGER des VAs. Cette thèse porte sur l'ensemble du processus, en partant de l'évaluation des risques, de la planification de la trajectoire jusqu'à la prise de décision et au contrôle du véhicule autonome. En premier lieu, une architecture multi-contrôleurs probabiliste (Probabilistic Multi-Controller Architecture P-MCA) est conçue pour une conduite autonome sûre en présence d'incertitudes. Cette architecture est composé de plusieurs modules interconnectés qui sont responsables de : l'évaluation du risque de collision avec tous les véhicules observés tout en considérant les prévisions de leurs trajectoires ; la planification des différentes manœuvres de conduite ; la prise de décision sur les actions les plus appropriées à réaliser ; le contrôle du mouvement du véhicule ; l'interruption en toute sécurité de la manœuvre engagée si nécessaire (en raison par exemple d'un changement soudain de l'environnement routier) ; et en dernier recours la planification des actions évasives à défaut d'un autre choix. L'évaluation des risques proposée est basée sur une stratégie à deux étapes. La première étape consiste à analyser la situation actuelle de conduite et à prévoir les éventuelles collisions. Cette étape est réalisée en tenant compte de plusieurs contraintes dynamiques et des conditions de circulation connues au moment de la planification. La deuxième étape est appliquée en temps-réel, durant la réalisation de la manœuvre, où un mécanisme de vérification de la sécurité est activé pour quantifier les risques et la criticité de la situation de conduite sur le temps restant pour réaliser la manœuvre. La stratégie décisionnelle est basée sur un réseau Bayésien de décision à niveaux séquentiels pour la sélection et la vérification des manœuvres (Sequential Decision Networks for Maneuver Selection and Verification SDN-MSV) et constitue un module essentiel de l'architecture P-MCA. Ce module est conçu pour gérer plusieurs manœuvres routières dans un environnement incertain. Il utilise l'évaluation des étapes de sécurité définies pour proposer des actions discrètes qui permettent de : réaliser des manœuvres appropriées dans une situation de trafic donnée, il fournit également une rétrospective de la sécurité, cette dernière actualise en temps-réel les mouvements de l'égo-véhicule en fonction de la dynamique de l'environnement, afin de faire face à toute situation dangereuse et risquée soudaine. Dans ce dernier cas, il est proposé de calculer le contrôle de bas niveau correspondant basé sur une stratégie d'évolution avec adaptation de matrice de covariance (Covariance Matrix Adaptation Evolution Strategies CMA-ES) qui permet à l'égo-véhicule de suivre la trajectoire d'évitement sans collision conseillée pour éviter un accident et garantir la sécurité à tout moment. La fiabilité et la flexibilité de l'ensemble de l'architecture P-MCA proposée et de ses composants élémentaires a été validé de manière intensive, d'une part dans des conditions de circulations simulées, avec différents scénarios de conduite, et d'autre part, en temps-réel avec les véhicules autonomes disponibles à l'Institut Pascal.

Mots-clés : Conduite autonome, Architecture multi-contrôleurs, Prise de décision dans l'incertain, Planification de trajectoire, Évaluation des risques, Vérification de la sécurité.

CONTENTS

LIST OF FIGURES

List of Tables

GLOSSARY

- **ACC:** Adaptive Cruise Control.
- **ADAS:** Advanced Driver Assistance Systems.
- **AIDP:** Actual Inter-Distance Profile.
- **ALC:** Automatic Lane Changing.
- **AVs:** Autonomous Vehicles.
- **BN:** Bayesian Network.
- **CMA-ES:** Covariance Matrix Adaptation Evolution Strategy.
- **CPT:** Conditional Probability Table.
- **DN:** Decision Network.
- **D-PLSB:** Dynamic Predicted Lower Safety Boundary.
- **DBN:** Dynamic Bayesian Network.
- **D-PIDP:** Dynamic Predicted Inter-Distance Profile.
- **EADL:** Evasive Action Decision Level.
- **ELC:** Elliptic Limit-Cycle.
- **ETTC:** Extended Time-To-Collision.
- **ITS:** Intelligent Transportation System.
- **KL-ACC:** Keep Lane with Adaptive Cruise Control.
- **LCL:** Lane Change Left.
- **LCR:** Lane Change Right.
- **LKA:** Lane Keeping Assist.
- **MCA:** Multi-Controller Architecture.
- **MDL:** Maneuver Decision Level.
- **MV:** Maintain Velocity.

- **OD-PAP:** Optimal Dynamic Predicted Angular profile.
- **OD-PIDP:** Optimal Dynamic Predicted Inter-Distance Profile.
- **P-MCA:** Probabilistic Multi-Controller Architecture.
- **S-PLSB:** Static Predicted Lower Safety Boundary.
- **S-PIDP:** Static Predicted Inter-Distance Profile.
- **SDN-MSV:** Sequential Decision Networks for Maneuver Selection and Verification.
- **SVDL:** Safety Verification Decision Level.
- **TTC:** Time-To-Collision.
- **V2V:** Vehicle-to-Vehicle.

1

GENERAL INTRODUCTION

Contents

1.1/ OVERVIEW

Transportation systems have provided to the humanity priceless social and financial advantages, they are also likewise connected with negative outcomes like traffic fatalities, gas emission, and traffic congestion. Today, like never before tech companies and research laboratories throughout the worldwide invest huge efforts to reduce these effects by automating the transportation systems. This technology has the ability to radically transform the transport sector and make roads much safer in particular by eliminating the human error and transcending its physical limitations (reaction time, bad visibility in adverse weather situations, etc.). However, after test vehicles suffered their first accidents, several institutions worldwide have raised concerns about guaranteeing safety in all conditions.

Over the past decades, the increase in the vehicles number on the road led to a sharp rise in the accidents number. For many years, the identification of accident cause has been the focus of intense research attention. These research reach a conclusion that 96% of accidents are due to human errors [224], for example when risk is not judged correctly and the driver reacts too slowly or incorrectly, with driver inattention being the main cause for vehicles accidents [232]. According to an estimation made by the United Nations (UN) [37], the number of fatalities worldwide will increase by almost 50% to reach 1.9 million from 2010 to 2020. Taking these statistics into account, the automotive industry has set itself the mission of reducing the accident number. Advanced Driver Assistance Systems (ADAS) came to help achieving this goal.
Bosch has conducted a study on ADAS in 2012 [37]. It shows that ADAS system such as Emergency Braking system, could prevent up to 72% of all rear-end collisions involving personal injuries in Germany. In addition, since the introduction of ADAS systems to prevent collisions, the number of accidents has fallen sharply on vehicles with this prevention

system: -5.1% for HondaAccord, -7.1% for Mercedes-Benz and -21% for Volvo [54]. Driver assistance systems includes:

- Driver Alertness Detection System (DADS) to prevent crashes caused by fatigue.
- Automatic Braking systems to prevent or reduce the severity of collision.
- Infrared night vision systems to increase seeing distance beyond headlamp range.
- Adaptive cruise control (ACC) which maintains a safe distance from the vehicle in front.
- Lane departure warning systems to alert the driver of an unintended departure from the intended lane of travel.
- Lane change assistance.
- Electronic Stability Control, which intervenes to avert an impending loss of control.
- Anti-lock braking systems (ABS).
- Emergency brake assist systems.
- Automated parking system.
- Obstacle detection sensor systems notify a driver how close their vehicle is to an object - usually providing a distance measurement, to the inch, as to how close they are.

The different levels of driving delegation to be illustrated by the corresponding driver assistance systems are shown in Figure 1.1. It summarizes the standard grading for vehicle automation as presented by the SAE International Standard 1 J3016 [57]. Nowadays, vehicles are quipped with ADAS from level 0 to 2 and they consist the first step toward the automation of driving functionalities to lead progressively toward fully autonomous vehicles. According to the scientific literature, the first automated vehicle was built in Japan in 1977, within the framework of the CACS (Comprehensive Automobile Traffic Control System) project. Under the supervision of Professor S. Tsugawa [205, 206], demonstrations were carried out with a vehicle capable navigating in lane on its own while using a camera that detects lane markings. The vehicle was successfully driven under various road environments at the speed within 30 Km/h.
An European research initiative proposed also one of the first autonomous cars developed by Ernst Dickmanns [65] in the 1988. This paved the way for new research projects, such as PROMETHEUS (PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety (1987-1995)), which aimed to develop a fully functional autonomous car. In 1994, the VaMP driverless car [64] resulting from the PROMETHEUS work managed to drive 1,600km, out of which 95% were driven autonomously. At a similar time, the CMU NAVLAB [119] was making advances in the area and in 1995 demonstrated further progress with a 5,000 km drive across the US out of which 98% was driven autonomously.

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**

Figure 1.1: Levels of driving automation by SAE International's new standard J3016 (SAE, 2014)

1.2/ Context and motivation of this PhD work

Driving is a complex task gathering strategic decision making, maneuver handling and controlling of the vehicle while accounting for external factors, traffic rules and hazard. The cognitive perception and response of the driver plays an important role in influencing the driving behavior and the vehicle control. The goal of an autonomous vehicle is to reach and exceed the driver perception of safety and judgment for dangerous situations and infer the driver's action. The core of a robust automotive safety system has the ability to handle the complexity of driving while having for instance appropriate partition for situation assessment method which defines the current driving state of safety while taking into account pre-planned trajectories, and a decision-making strategy that makes the control decision. Safety in the domain of autonomous vehicles denotes the ability to respect traffic rules and avoid potential collision with other traffic participants.
One of the major research topics in this domain, is enabling vehicles to cope with any environment traffic condition while making the appropriate decision and guaranteeing the safety of maneuvers even in presence of uncertainty. Although multiple ADAS have successfully improved safety, fatal car crashes still occur. This is mainly caused by measurement uncertainties and unexpected maneuvers of other traffic participants. For this reason validating the safety of self-driving vehicles while applying safety verification methods can prove the coherence of the vehicles' behavior, reduce remaining risks and the need for extensive testing and more importantly allow us to plan evasive maneuver, in real-time.

In this context, our research work has been launched between Sherpa Engineering and the Institut Pascal in order to study the safety of trajectories taken by autonomous vehicles. We aim at designing a multi-controller architecture that combines: the safety assessment and verification, the probabilistic decision making process, the trajectory planning, and the control law part.

1.3/ PROBLEM FORMALIZATION

Decision making for autonomous driving according to [194, 207] requires: *"rapidity"* in at least the planned decision for the driving maneuvers; *"coherency"* as the decision making module should be consistent avoiding unnecessary switch in the plan; *"providentness"* meaning the module should foresee how the situation will evolve after some time or some maneuvers and include it in the decision making; and *"predictability"* as the decision making should act while conforming to a human driver perception of safety and judgment for any situation.

The purpose of this thesis is to develop the necessary systems while accounting for the aforementioned characteristics and be able to:

1. **Assess the risk in the surrounding environment** by defining appropriate risk metrics to discriminate between different alternatives of decisions. This implies the prediction of the short-term behavior of the surrounding vehicles (3 to 5 seconds) and the prediction of the infrastructure components for example the lane marking [97] and the traffic scene.
2. **Take appropriate decision in nominal driving situation** Using the aforementioned elements to determine the most suitable maneuver for the vehicle to execute.
3. **Execute the decided maneuver** By using motion planning and control algorithms.
4. **Verify the coherence of the executed maneuver** While using a dedicated safety verification module that quantifies the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver. This module provides a safety retrospection that updates in real-time the ego-vehicle movements according to the environment dynamic.
5. **Plan evasive maneuvers** In case of unexpected behavior or failure of the perception system detected thanks to the safety verification module present in the decision maker module, we resort to the planning of evasive maneuvers and guarantee the *coherency* principle defined above.

The following flowchart 1.2 clarifies the above characteristics of a decision-making system motivated and applied in our PhD thesis work.
However, despite the investment of researchers and tech companies in fully autonomous vehicles, the technology doesn't seem ready yet to be deployed on road and we can identify many challenges:

- **Reliable perception/localization systems in any situation** The faultless understanding of the road scene even in worst case scenarios such as bad weather. Indeed, a miss detection of a traffic participant will lead to an accident.

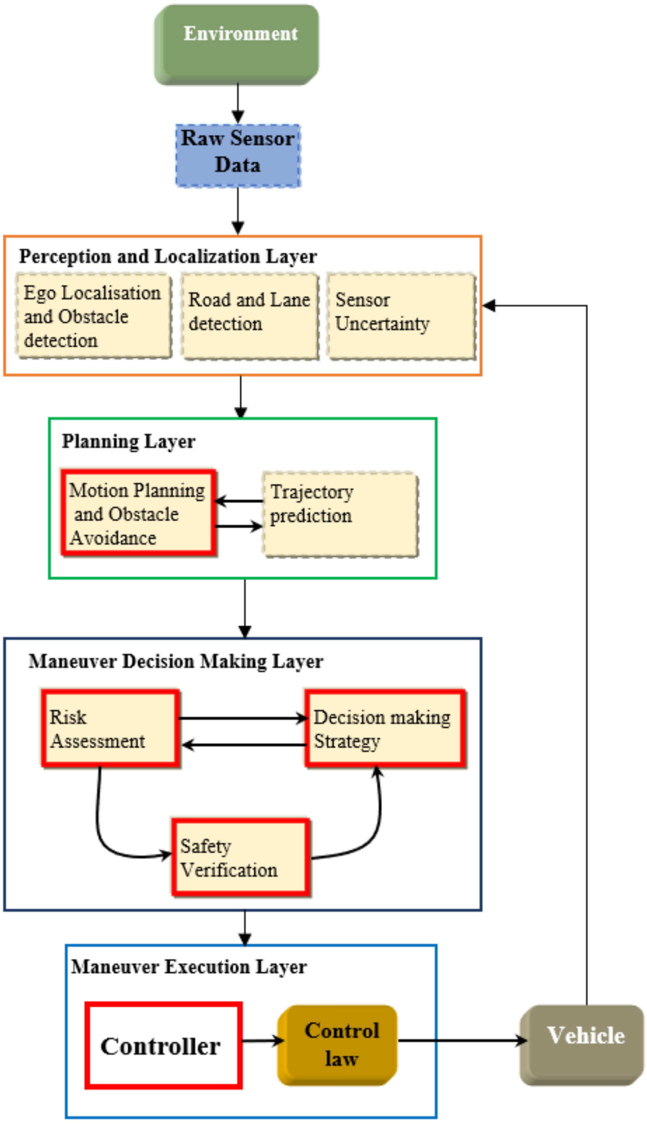


Figure 1.2: Simplified flowchart of a decision making architecture for autonomous vehicles navigation. The blocks with thicker red boxes are the focus of this dissertation.

- **Standardized metrics to define safety and the ability to validate and generalize the system to all cases** There is a lack of formal analysis and evaluation methods to assess performance. However, a host of new artificial intelligence (AI) technology standardization programs will be starting out in 2020 [113]. Their common goal is to establish safety standards in autonomous vehicles (AV) and robotics for AI systems.

- **How to obtain the right balance between respecting the traffic rules, the comfort of the passenger and the safety of the vehicle** An important challenge in this field is to find the perfect balance between ensuring safety with all the imposed constraints: uncertainties, complexity and not being too conservative in the navigation.

- **User acceptance and Ethical issues** As soon as robots (or any embedded artificial system) interact with humans, ethical question arise. Many approaches have been studied in literature to propose ethical decision making strategies [86], especially in critical situations where for example morality and deontology [200] constrain the system to act based on the determination of costs.

In this Ph.D work, we aim at proposing system architecture unified, generalized, modular. The purpose is to be able to derive appropriate decision maneuver in nominal driving and also to guarantee safety of navigation in emergency situations. In addition, a particular focus will be given to study the lane change maneuver mainly in highway. This task is considered as the most useful and complex task [137] to perform mainly if the surrounding vehicles are very dynamic and in certain cases unpredictable.

TARGETED SCENARIOS

In a traffic situation, the ego vehicle has to adapt its behavior to the surrounding traffic environment. To pursue a lane change maneuver for example, the ego-vehicle has to position itself in a gap between traveling in the target lane while ensuring a safe distance between the nearby vehicle in its current lane during all the maneuver. Thus, to have safe and feasible lane change maneuver, this one should be planned such that we respect the followings:

- Avoid potential collision with other traffic participants.
- Respect traffic regulations.
- Satisfy the dynamic constraint and limitations of the vehicle.
- Respect the comfort of passengers with reduced level of lateral and longitudinal acceleration along with jerking.

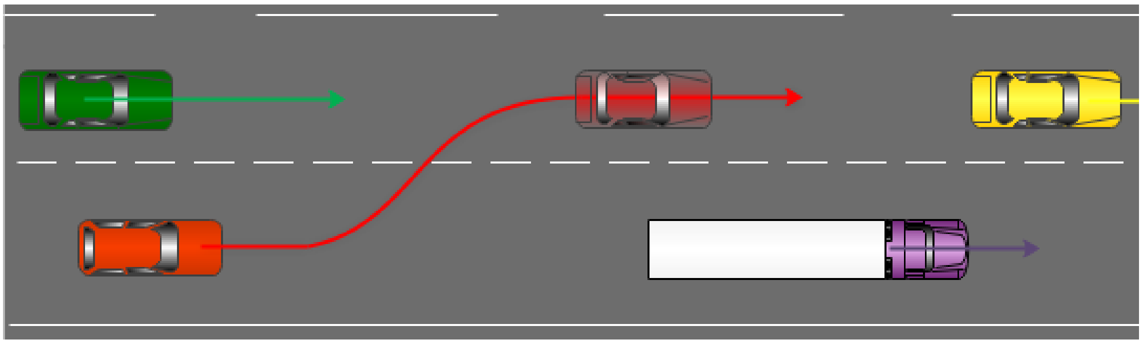
In general, as discussed in [164] in addition for a lane change maneuver to be safe and feasible it should be desirable and advisable by the system in three conditions:

- Improve the driving conditions and maintain a desired velocity by overtaking a slow vehicle.
- Allow a rear faster vehicle to pass by changing lane to the right and let it pass in safety.
- Avoid an imminent collision or a lane drop.

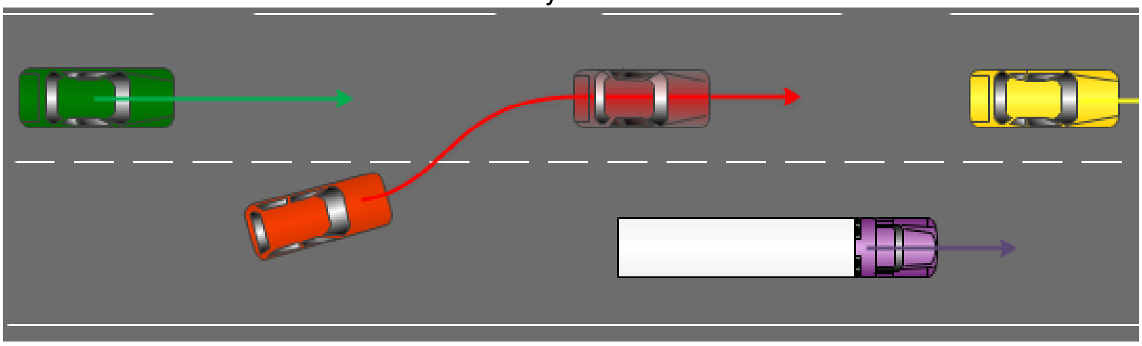
The two first conditions will be defined in this dissertation as nominal driving situations. The third condition will be defined as emergency situations. In all of these conditions, the decision making system should be able to predict and react to this change and guarantee safety in all case. Theses can be summarized in two scenarios illustrated in Figure 1.3.

1.4/ PROPOSED APPROACH AND CONTRIBUTIONS

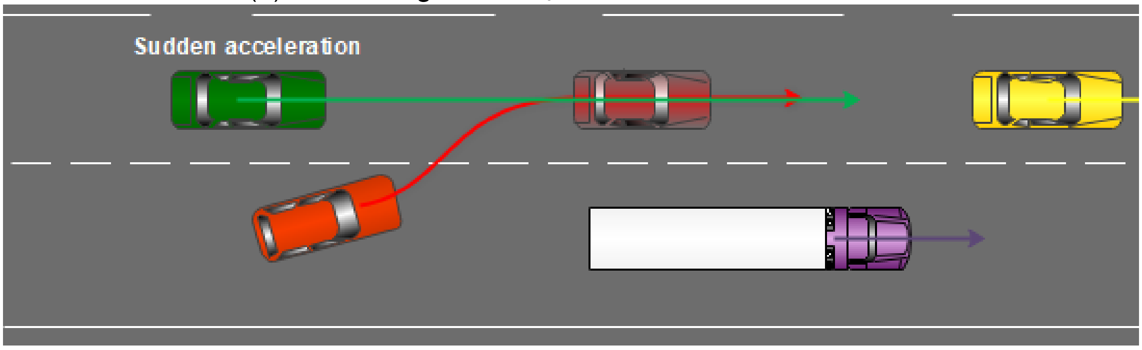
This thesis presents a probabilistic framework for handling navigation of autonomous vehicles in an end-to-end manner through a Probabilistic Multi-Controller Architecture (P-MCA) that is responsible for:



(a) The first phase of an overtaking maneuver at time t_0. The planned ego vehicle path is illustrated by the red arrow.



(b) Overtaking at time $t_0 + \Delta t$ in nominal conditions



(c) Overtaking at time $t_0 + \Delta t$ in emergency situations. The vehicle green suddenly and strongly accelerates during a lane change maneuver of the ego vehicle.

Figure 1.3: Vehicles traveling on a two lanes road. The ego vehicle in red performing a lane change maneuver on a slow truck in front. The thick (red, green and purple) arrows represent the predicted path and the planned route for the vehicles.

- Assessing the collision risk with all observed vehicles while considering their trajectories' predictions,
- Planning the different driving maneuvers,
- Making the decision on the most suitable actions to achieve,
- Control the vehicle movement,
- Aborting safely the engaged maneuver if necessary (due for instance to a sudden change in the environment) and as last resort planning evasive actions if there is no other choice.

In this regard, the main contributions in the presented dissertation are:

- **Obstacle avoidance strategy for lane change maneuvers** It is proposed (cf. Chapter 5) the adaptation of the existing work on limit-cycle trajectories at the Institut Pascal [9, 11] from mobile robot navigation in cluttered environment to road-way for autonomous vehicle, that will constitutes the baseline of the used obstacle avoidance strategy.

- **A hybrid Multi-Controller Architecture (MCA)** The design of an overall multi-controller architecture (cf. Chapter 5) that effectively links probabilistic decision-making with its risk assessment and management modules, trajectory planning and control for safe navigation of autonomous vehicles for an effective and safe navigation of autonomous vehicles. Hybrid denotes the combination of model-based and artificial intelligence approaches.

- **A novel formulation of risk** The proposed risk assessment (cf. Section 5.3) is based on a dual-safety stage strategy. The first stage analyzes the actual driving situation and predicts potential collisions. This is performed while taking into consideration several dynamic constraints and traffic conditions that are known at the time of planning. The second stage is applied in real-time, during the maneuver achievement, where a safety verification mechanism is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver.

- **A probabilistic decision making framework** Proposition of a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) (cf. Chapter 6) designed to manage several road maneuvers under uncertainties. It utilizes the defined safety stages assessment to propose discrete actions that allow to: derive appropriate maneuvers in a given traffic situation and provide a safety retrospection that updates in real-time the ego-vehicle movements according to the environment dynamic, in order to face any sudden hazardous and risky situation. This approach constitutes a good deal for handling numerous scenarios configuration, multiple decision criteria while taking uncertainty into account. The graphical representation of the Decision Network (DN) eases the connection between the situation assessment level (observations level) using the safety measures and the decision-making level.

- **An optimal evasive strategy** It is proposed to compute the corresponding low-level control (cf. Section 6.4) based on an evolutionary algorithm (EA), the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) that allows the ego-vehicle to pursue the advised collision-free evasive trajectory to avert an accident and to guarantee the safer action to apply at any time. Indeed, this will allow the overall system to increase its degrees of freedom concerning the maneuverability of the vehicle and allow smooth changes during the evasive actions and thus ensure the safety of the system and comfort of the passengers.

- **Implementation of the Probabilistic Multi-Controller Architecture** The reliability and the flexibility of the overall proposed P-MCA and its elementary components have been intensively validated (cf. Chapter 7), first in simulated traffic conditions, with various driving scenarios, and secondly, in real-time with the autonomous vehicles available at Institut Pascal.

1.5/ MANUSCRIPT OUTLINE

In addition to the global introduction and final conclusions and prospects, the manuscript is composed of 7 chapters which are decomposed into two parts.

The first part titled ***"State of the art"*** corresponds to the state of the art and bibliographical review on the different used control architectures and on two major building blocks composting these architectures namely: the risk management and the decision-making strategies for Autonomous Vehicles (AVs). The goal through this first part is to lead the reader through the process, concepts and methods defining a complete overall architecture for autonomous vehicles able to ensure high level of safety. This part is composed of the following chapters:

- **Chapter 2 - Control architectures for autonomous vehicles** presents related work on the different used architectures in the literature that represent the backbone combining the different modules of an autonomous vehicles: perception, localization, decision making, motion planning and control. A classification is proposed for each part of the different existing approaches with an emphasis on methods that have shaped the history of AVs.
- **Chapter 3 - Safety assessment and management of AVs** This chapter presents the related work on risk estimation and safety verification techniques. A classification is proposed of the different existing approaches.
- **Chapter 4 - Decision-making for AVs** This chapter is dedicated to the related work on decision making. A classification is proposed of the different existing approaches.

The second part titled ***"Proposed multi-controller architecture for safe autonomous navigation under uncertainties"*** corresponds to the contributions proposed during this PhD thesis namely: the Probabilistic Multi-Controller Architecture (P-MCA), the risk assessment and management strategy, and the decision-making framework. This part contains three chapters:

- **Chapter 5 - Probabilistic Multi-Controller Architecture (P-MCA)** It is detailed in this chapter the composition of the proposed architecture that combines the motion prediction and path planning, the threat assessment, the decision-making and the control algorithms. Details are first given about the main assumption and background work. Then an introduction of the principles and the main functionalities behind the proposed P-MCA are detailed with a focus on the motion planning modules for road-way navigation. The rest of the chapter is dedicated to the risk management strategy and to the proposed criteria for anomaly detection. Throughout the chapter demonstrative examples are shown to validate each part.
- **Chapter 6 - Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV)** Once appropriate risk metric are available, the proposed decision strategy is detailed in this chapter. Firstly, the Bayesian formalism for decision making is described. Then, each level of the decision network and its interacting variables are explained in the first sections. Afterwards, is detailed the evasive maneuver strategy based on the CMA-ES algorithm. Finally, various simulation were presented in different traffic situations to validate the decision-making framework

- **Chapter 7 - Implementation of the Probabilistic Multi-Controller Architecture**
  This chapter describes the performed experiments and the used tools for the implementation of the Probabilistic Multi-Controller Architecture (P-MCA) for autonomous navigation.

I

State of the art

2

CONTROL ARCHITECTURES FOR AUTONOMOUS VEHICLES

Contents

This chapter presents related work on the different kinds of motion control architectures proposed in the literature. These architectures represent the backbone combining the different modules of an autonomous vehicles: perception, localization, decision making, motion planning and control. A classification is proposed for each part of the different existing approaches with an emphasis on methods that have shaped the history of AVs.

2.1/ INTRODUCTION

In recent years, researches and tech companies have been racing towards self-driving cars and numerous safety systems have been among the main focus of the automotive research area.

In the field of automotive safety the words “active” and “passive” are essential concepts. “active safety” is used to refer to technology that helps prevent accidents and “passive safety” of vehicle components (primarily airbags, seat belts and the physical structure of the vehicle) that help protect occupants during an accident. In this Ph.D work our interest goes towards active safety systems where the main challenge lies in being able to reduce the risk, improve comfort and enhance the global vehicle’s performance. Some of these active safety systems are already commercialized. They are commonly called Driver Assistance Systems (DAS) (cf. Chapter 1) and they greatly increase the safety of a vehicle. Those functions constitute the first steps toward autonomous vehicles and they represent Level 1 in the SAE standard grading for vehicle automation [57] (cf. Figure 1.1). Many approaches have been proposed in the literature to classify control architectures either using model-based [134, 173, 191] or data-driven [101, 203, 221] approach. Some other works try to make a classification [115] of the tasks that an autonomous vehicle may do and its modeling from the data-driver perspectives and the classical methods to design these decision-making architectures. Figure 2.1 shows the basic components of an autonomous vehicle from the classical or the end-to-end perspectives that will be presented in the following sections.

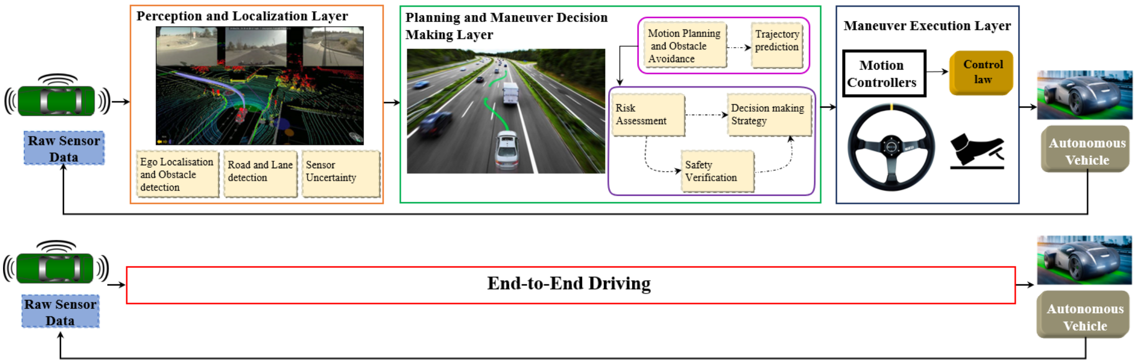


Figure 2.1: Standard components in an autonomous driving systems listing the various tasks from the classical and the end-to-end perspective.

Nevertheless, the common goal of all of theses automotive safety architectures is to process the stream of information from proprioceptive and exteroceptive sensors. These observations together with the prior knowledge of the road architecture, traffic rules, vehicle dynamics are used to assess the actual risk, deduce the most suitable decision and automatically select the appropriate controller to achieve desired actions thought the low level control variables which are used to induce the vehicle dynamic.
In this chapter we will focus on control architectures that belongs to level 3 according to the SAE [57] (cf. Figure 1.1) as we aim to analyze the problematic of safety as an integrated system approach that is able to account for all the vehicle driving states and where each sub-system interact with the other.

2.2/ Definitions

Model-based denotes the use of mathematical representation in the modeling of the system and thus incorporates a physical understanding of the system. Based on this understanding, vehicle's motions and the uncertainty evolution is formalized analytically in the model-based control architecture. Its main field are: system identification, adaptive control, robust control, optimal control, variable structure control, Lyapunov-based controller designs [215] have been extensively used in the literature. These methods often use statistical estimation techniques e.g., Kalman Filtering or Particle Filtering to estimate uncertainty. The architecture of model-based control theory is shown in Figure 2.2. This diagram shows that the system model and assumptions are the starting point of the architecture design, but also the destination of the model-based control system output in a kind of feedback. Data-driven in the other hand is preferred when the system models is not available or hard to model but instead the system data and historical properties are available. In fact, nowadays a huge amount of process data is stored at every time instant. These data contain all the important state information. Using these data, on-

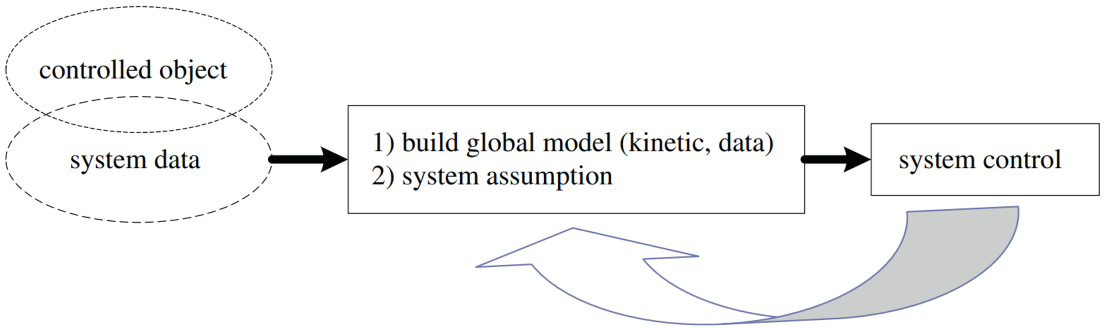


Figure 2.2: Simplified architecture of a model-based control architecture (Image credit: [103])

line and off-line, to design controller, assess risk, predict trajectories or make decision become very relevant especially under the lack of accurate models. Machine learning approaches, neural networks, deep learning are illustrative of this approach. However, these techniques require a huge diversified database for the learning phase. Indeed, in order to have algorithms that can be generalized the database have to cover all the driving scenarios which can not be possible in real world application. What we defined in this dissertation as “Hybrid”, is architectures or techniques that combines both the classical concepts and data-driven formalism (cf. Chapters 5 and 6)

In this chapter, we propose a state of the art of the different configurations of these architectures shown in the literature and the interactions between the modules especially the motion planning, the risk assessment and the decision-making, while highlighting their importance in ensuring safety. Following the taxonomy used in [52] (cf. Figure 2.3), three paradigms emerge for autonomous driving systems:

- *Mediated perception approaches* in the other hand analyze the entire scene before making a driving decision. It involves multiple sub-components: lane detection, cars and pedestrian recognition, etc. The results are combined and then given to upper layers to take the driving decisions. Most nowadays autonomous driving are based on this last paradigm and it is categorized in this manuscript as classical methods (cf. Section 2.3).
- *Behavior reflex approaches* that maps directly an input image for example to a driving action like steering, and this is what we refer to in this dissertation as end-to-end driving (cf. Section 2.4).
- The third paradigm is called *direct perception* approach and falls in between the two previous paradigms and allow the right level of abstraction. This will be discussed in the hybrid methods in Section 2.5.

2.3/ Classical system architecture for autonomous vehicles

The classical control architecture consist to organize and partition the problem of automating the driving task into a multitude of parts: localization, perception, motion plan-

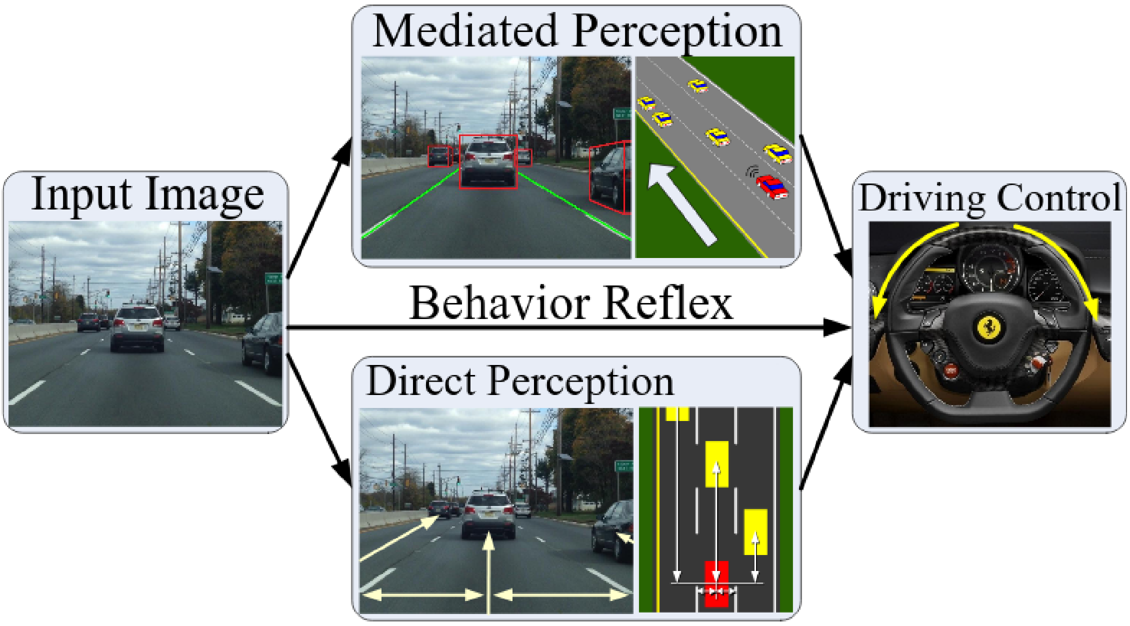


Figure 2.3: Three paradigms for autonomous driving (Image credit: [52])

ning, decision-making (also called the behavior generation/behavioral layer) and control. Each of these parts is divided in a multitude of sub-tasks. For example the decision making module can be partitioned as a situation assessment method which defines the current driving state of safety while taking into account pre-planned trajectories, and a decision-making strategy that makes the control decision. The path planning on the other hand can be partitioned in trajectory prediction, obstacle avoidance, path following, behavior generation. All of this behaviors are organized in a hierarchical structure able to handle the coordination while guaranteeing the stability of this systems.

The Defense Advanced Research Projects Agency (DARPA) Grand Challenges gave a new impulse to the research in autonomous vehicles and on the design of complex system architecture to autonomous driving. The major challenge in comparison to previous demonstrations in that there was no human intervention during all the race. Whereas the 2004 and 2005 DARPA Grand Challenges were intended to demonstrate that autonomous vehicles can travel significant distances, the 2007 DARPA Urban Challenge (DUC) was designed to promote and encourage innovations in autonomous vehicles in cluttered urban environments. We will details in what follows the overall architecture of each edition winning team. No winning team was declared in the first DARPA edition because non of the robot vehicles finished the race. However, a vehicle named Sandstorm went the farthest and gave a hint on the autonomous vehicles capabilities providing a hint on how to win the challenge [209]. Therefore, a second DARPA Grand Challenge event was scheduled for 2005.

The racing team from Stanford University won the second edition of the DARPA Grand Challenge with the robot Stanley. The winner of the race had to complete the course in less than 10 hours. The Stanford racing team had the best time with 6 hours and 53 minutes. Sebastian Thrun and his team published afterwards a paper that described the main components of the winning architecture [202] and released the source code. Their main challenge was in the perception systems for road finding and obstacle detection, as well as high-speed obstacle avoidance (cf. Figure 2.4). The architecture shown in Figure 2.4 is divided into six main functional groups: sensor interface, perception, control, vehicle interface, and user interface. This architecture is related to the well known three layer architecture proposed by Erann Ga in 1998 [78] though without a long term planning. It directly uses the information form perception to generate the feasible trajectory based on the estimated state of the vehicle and the detected obstacles. Following the description and data of the course provided by DARPA, no global path planning was needed and only a local planner for obstacle avoidance was sufficient. Stanley was moved left or right by changing a lateral offset with respect to the base trajectory. The objective of the proposed planner is to compete the course as fast as possible while avoiding obstacles and staying inside the boundaries of the road.

From a general point of view, Stanley's software reflects common approach in autonomous vehicle architecture design that is why in this dissertation it was cited in the classical methods section. Nonetheless, many of the individual modules relied on state-of-the-art of Artificial Intelligence techniques. The use of machine learning, both ahead and during the race, made Stanley robust and precise. This was a defining moment in self-driving cars development, recognizing Machine Learning and AI as central components of autonomous driving. The defining moment is additionally eminent and notable, since most of the literature work in this domain is dated after 2005.

While these two first editions of the DARPA Grand challenge was a milestone in the quest for self-driving cars, it left a number of open questions and challenges. According to the winner [202], Stanley is not able to navigate in traffic and this is exactly the subject of the next DARPA Challenge.

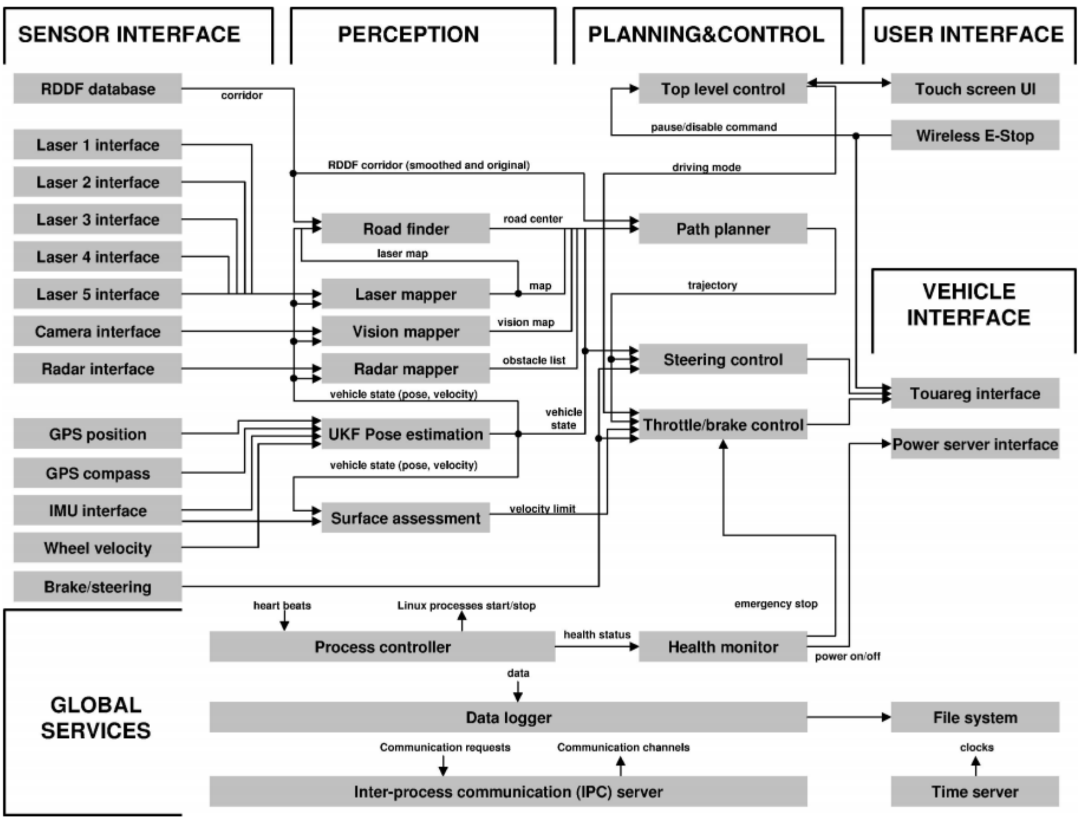


Figure 2.4: Flowchart of Stanley Software System (Image credit: [202])

As mentioned above, the 3^{rd} DARPA Grand Challenge also known as the "Urban Challenge" happened in 2007 and dealt with navigation in urban environment. The objective was to prove that it could be done in traffic when there were both moving intelligent vehicles and moving vehicles driven by humans. Competitors had to drive 97 km through urban environments, interact with other traffic participants and obey to the California traffic rules. This made it necessary to the competing team to use decision-making mechanism to guide the behavior of the controlled vehicles. Many teams participating in the challenge chose common architecture design by creating modular architecture with a variety of interconnected modules. Concerning the path planner, different algorithm have been employed such as Rapidly-exploring Randomized Trees (RRT). The control part used a wide range of control algorithms such as sampling or feedback based [45]. The most common approach in this competition was to use Finite State Machine (FSM) as the way to represent the decision-making strategy [131]. As an example, Junior and Odin (the second and third ranked team in the competition) used FSM to govern their vehicles' behavior. The FSM is given by a finite set of states in which the agent can be, and by the transitions between the states in response to some inputs. Based on the state of the vehicle an the world, the FSM searches for a suitable behavior (changing lane, making a U-turn) that makes the robot reach an objective checkpoint. Then, it either sends the controller the chosen trajectory that in return send steering and velocity command or send a message to the planner that the checkpoint cannot be reached. Most of these transitions are coded and tested by hand and thus prone to errors. Some team have used formal safety verification techniques to validate some parts of their architectures but the majority of them were tested in simulation without any formal guarantee that it would do what the designer intended it to do.

The Tartan Racing Team which was composed of staff from different entities including Carnegie Mellon University, General Motors, Caterpillar, Continental, and Intel, finished first in 250 minutes and 20 seconds with their vehicle *Boss*. *Boss* was capable of driving safely in traffic at speeds up to 48 km/h. The system architecture (shown in Figure 2.5) created for the race generated many innovations and where summarized in publications and several books [45, 208, 210]:

- Perception modules able to detect and track moving and static obstacle.
- Road localization and shape estimation to drive in whether a prior on the road geometry is available or not.
- Mission planning that computes the cost of all possible routes. Its objective is to attain a particular checkpoint in an optimal manner which means that it accounts on prior knowledge of congestion, legal speed limit, distance like a human would do.
- A behavioral module (shown in Figure 4.1) that acts as a tactical decision-maker, to execute the mission plan and handle errors and failure when they occur, is able to follow the traffic rules and violate them if necessary [210].
- A error recovery system that is able to detect and address errors from the motion planner and other aberrant situations and be able to generate novel sequence of goals and increasingly risky maneuvers as time progress while ensuring minimal complexity. However, even though Boss could recover from many failures, it took considerable time and human intervention would have reduced the overall race time by 15 minutes.

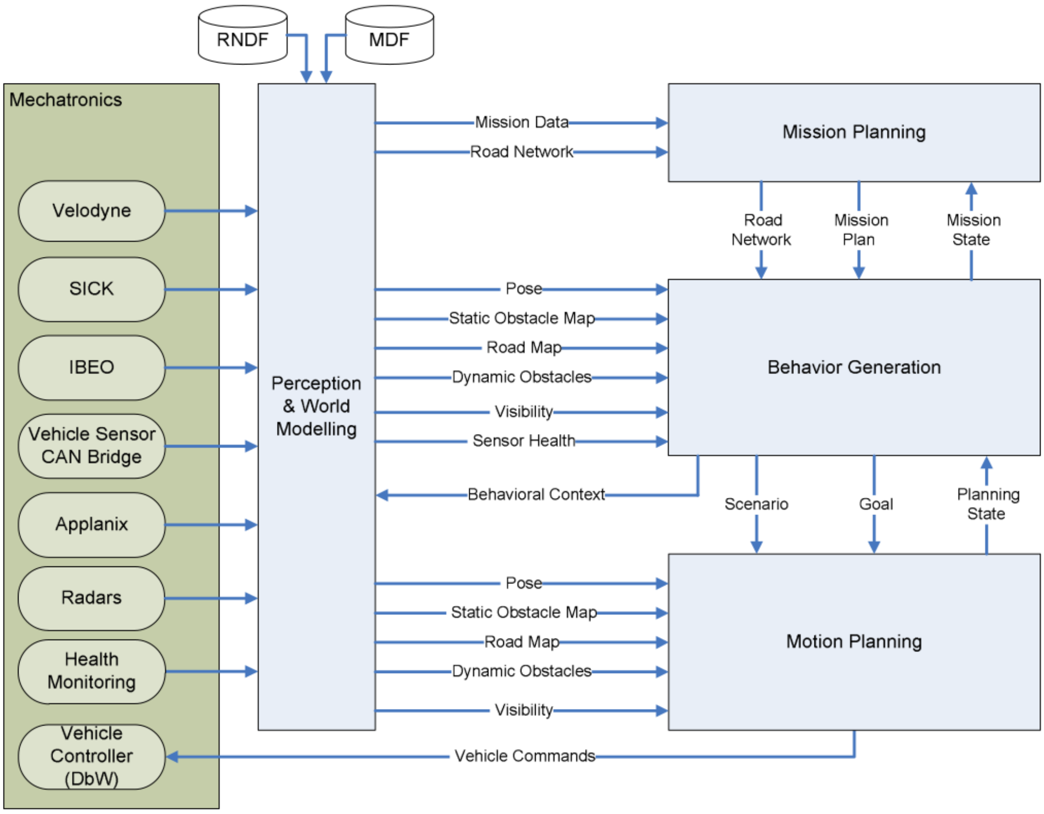


Figure 2.5: Boss's Overall Architecture (Image credit: [210]). Composed into: Mission Planning, Motion Planning, Behavior Generation, Perception and World Modeling, and Mechatronics. The RNDF and the MDF are respectively the Road Network Definition File and the Mission Data File

- A mixed-mode planning system able to navigate on road and safely maneuver through urban areas or parking lots.

Umrson et al., in [211] stated that despite the strength of the Urban Challenge participants in dealing with complex scenarios, their vehicles poorly operate around pedestrians and all relied on expensive sensors. In addition, in any complex system of this span, the assumptions of the designers will always be violated by unpredictable situations. Whereby, necessitating the development and verification of more principles and robustify the approach to recover from mistakes, which is an important issue for robotics research. In addition, integrating autonomous vehicles into our society with human drivers and pedestrians is a challenging problem as the autonomous cars need to be able to interpret hand gestures and eyes contact in situations where the traffic rules are violated or understand the willingness of other vehicles to change lane by them varying their speed making the gap between themselves and the other vehicle smaller. The learned lessons from the Urban Challenge were very valuable and a lot of them are still subject of nowadays research.

Numerous major competitions have been carried out since the DARPA challenges. Worthy examples to state include Hyundai Autonomous Challenge in 2010 [49], the VisLab Intercontinental Autonomous Challenge in 2010 [43], the Grand Cooperative Driving Challenge 2011 (GCDC) was the first international competition to implement highway platooning scenarios of connected cooperating vehicles [79]. Simultaneously research both in academia and industry pursued their way at an accelerated rhythm. The Google self-driving car, Tesla Autopilot system, the Uber's self-driving car are examples of commercial impressive work which expands the expertise acquired from the Urban Challenge.

Another pioneering event in the automotive history is the Bertha-Benz historic route. In august 1888, Bertha drove her husband's *Benz Patentmotorwagen Number 3* without him knowing from Mannheim to Pforzheim, Germany over 100Km. The popular responses on her journey paved the way for the economic success of her husband and the adoption of the automobile in the society. Exactly 125 years later, a collaboration of Daimler AG and KIT (Karlsruher Institut für Technologie) automated a Mercedes-Benz S-Class named "Bertha" that repeated the Bertha Benz Memorial Route but this time in a complete autonomy [75, 235, 236, 237]. The Bertha-Benz historic route is particularly challenging as it covers rural roads, 23 small villages and major cities and passes a different variety of traffic scenarios, narrow streets, intersections and roundabouts with oncoming traffics. The autonomous vehicle had to react on a variety of objects: parked cars, bicycles, pedestrians.
The system architecture proposed by [236] for this purpose is shown in Figure 2.6. The

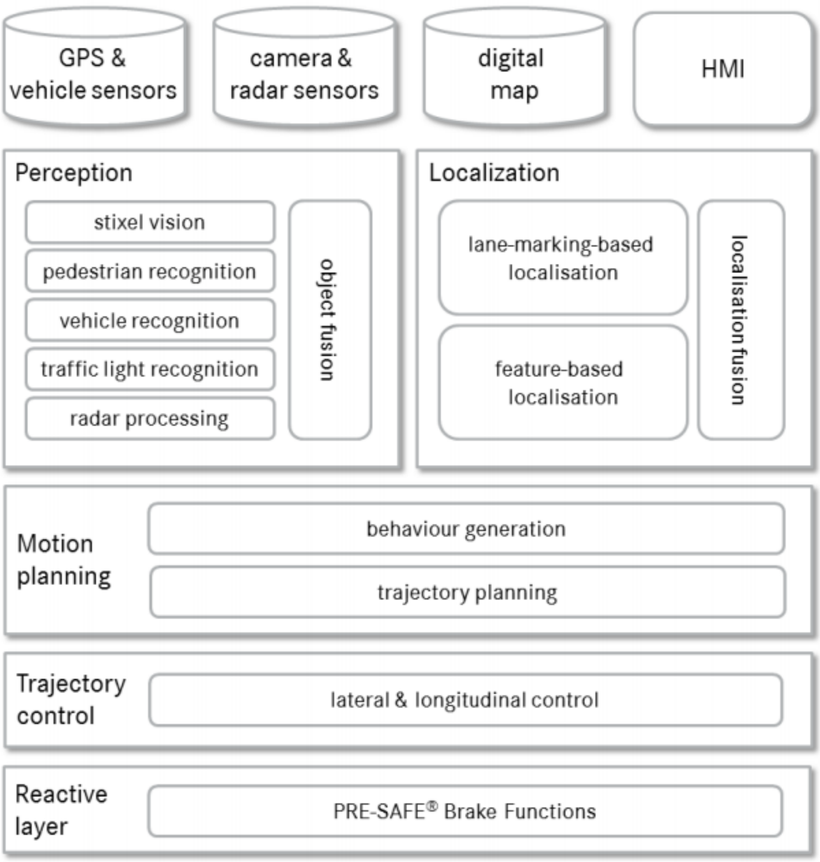


Figure 2.6: System overview of the Bertha Benz experimental vehicle (Image credit: [236])

architecture follows the classical structure by layer [236]. The perception and localization modules comes first and feed the lower layers with the processed perceptive information then an optimal trajectory generation is performed based on a continuous optimization.

The solution trajectory [236] is the constrained extremum of an objective function that is designed based on the dynamic feasibility and comfort. Static and dynamic obstacle constraints are incorporated in the optimization in a form of polygons. The constraints are designed in such away that the solution converges to a single, global optimum. The trajectory is then transformed into actuator commands by lateral and longitudinal controllers.
A reactive layer for emergency braking already existing in the Mercedes Benz S-Class 500 INTELLIGENT DRIVE is activated in the vehicle and support the autonomous driving functions in such a way that emergency breaking is not considered in the trajectory planning and control modules. At the end of this journey, the team working on the project stated that the overall car behavior is still far inferior then the performance of on a attentive human and one way to achieve comparable behavior is to improve the ability of the vehicle to interpret a given traffic scenario and predict the behavior of other traffic participants.

Another type worth stating architecture design is the multi-controller architecture. It is a kind of control architectures that allow to address complex task in a modular bottom-up way [8, 215]. They are referred in the literature as behavioral architecture and was introduced by Brooks in 1986 [21, 44]. They are based on the concept that a robot can achieve a global complex task while using a coordination of several elementary behaviors. The parallel in control is to decompose the global control into a set of elementary controllers like ADAS: Lane Keep assist, Automatic Lane Change, Adaptive Cruise Control in the purpose of mastering the overall robot/autonomous car behavior. The development of a multi-controller architecture pass inevitably by mastering the flux of commands [10] generated by the multitude of elementary behaviors/controllers composing the control structure. These elementary behaviors can be selected or processed to generate the desired action according to a coordination mechanism: the action selection. The design is based on hierarchy in which one behavior has the highest priority w.r.t. the others. The resulting actuators commands are given by the active controller with the highest level of priority at each step time. The main advantages of multi-controller architectures arise from their bottom-up construction and flexibility to deal with several complex tasks. Nevertheless, the challenging issue of this kind of architecture is to prove its reliability and stability while introducing mathematical analysis for each developed controller, as well as for the overall multi-controller architecture. According to [8] the challenging issues that should be addressed are the following:

- For elementary controllers, they must be stable and reliable for different environment contexts.
- For the coordination mechanism between controllers:
  - Master the coordination between controllers' actions in order to achieve safely and efficiently the assigned task,
  - Avoid jerking/discontinuities in term of the robot's control. The objective is to obtain stable and smooth switching between controllers.

Such a behavior-based architecture and action-selection mechanism have been used by the team VictorTango that arrived third in the DARPA Urban Challenge 2007 with the autonomous car ODIN. In Figure 2.7, the chosen Action Selection Mechanism operates within the Behavior Integrator that chooses the winning driver depending on the current situation.

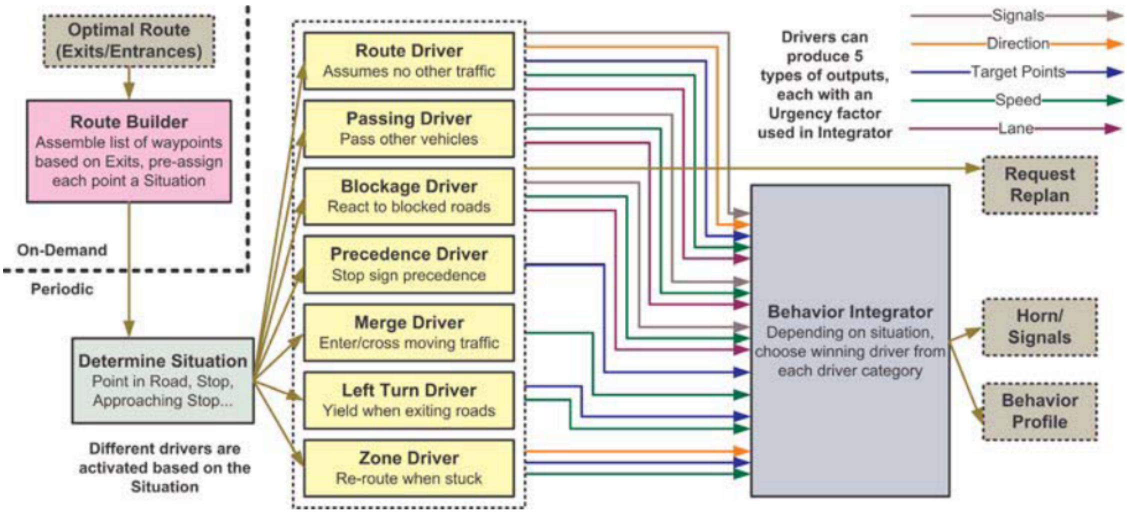


Figure 2.7: Flow diagram of the Behavior-Based architecture and the Action-Selection Mechanism (Image credit: [45])

Another architecture that illustrates the functioning of a multi-controller architecture is shown in Figure 2.8 and have been proposed by [9]. The architecture propose to ally long-term (with quasi-full knowledge on the environment) and short-term planning (navigation is done with the minimum information on the environment) with trajectories based on Parallel elliptic limit-cycles (PELC), to achieve navigation that can be qualified respectively as cognitive or reactive.

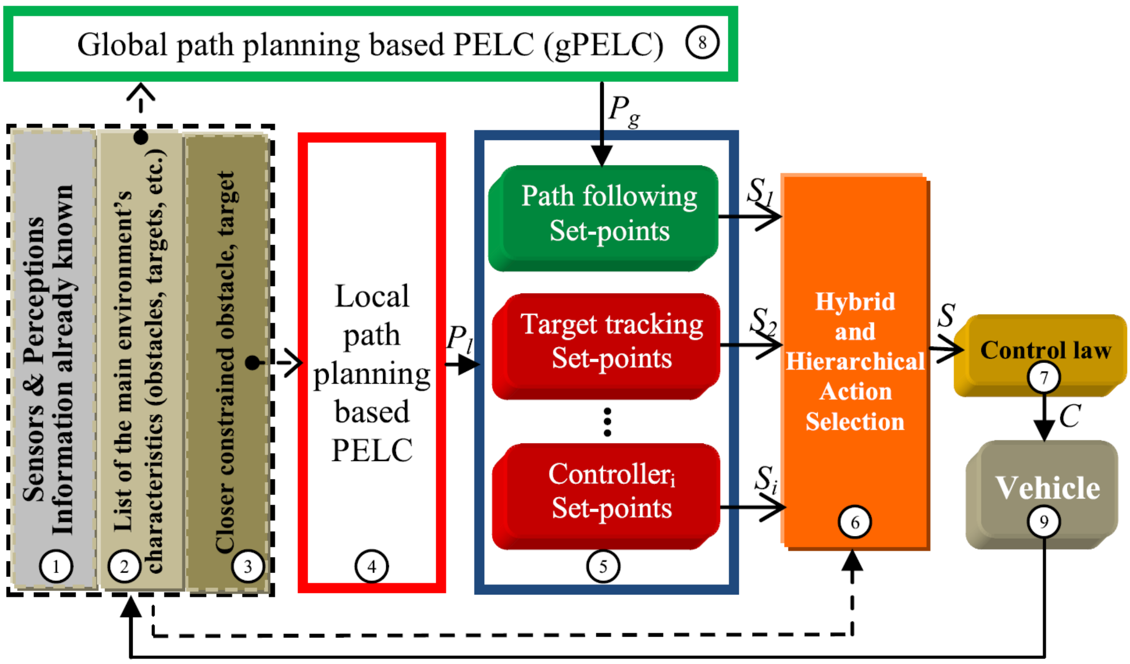


Figure 2.8: The cognitive/reactive multi-controller architecture (Image credit: [9])

Multiple other control architectures have been surveyed in the Ph.D. thesis of Vilca [218, 219].

Even though, the classical architectures design are the most used in the autonomous driving industry, the total scene understanding that is required by these architectures composition to derive a decision may add unnecessary complexity to the overall system when in most situation only a small portion of the detected objects are indeed relevant. In addition, the individual sub-tasks involved in each of the modules of theses architectures are themselves subject to open research.

2.4/ END-TO-END AUTONOMOUS DRIVING

Instead of keeping all the modules composing the automotive system architecture separate, an alternative framework proposes what is called "end-to-end" method, which integrates perception, localization, planning and decision making, that generate a control input for the vehicle. Typically such methods mostly rely on machine learning.

The *behavior reflex* approach dates back to the ALVINN (Autonomous Land Vehicle in a Neural Network) system proposed by Dean Pomerleau with the CMU NavLa pioneered end-to-end driving in 1989 [177] by teaching an artificial neural network to output steering angle to keep the vehicle on the road by taking images from a camera and a laser range finder as inputs. By 2006, it was at that point conceivable to learn how to avoid obstacles directly from raw stereo-camera inputs and that was achieved by DAVE (DARPA Autonomous VEhicle) [159]. It is trained to predict the steering angles from data provided by a human driver during training. The collected data gather a wide variety of terrains, weather condition and obstacle types. The learning system is a large 6-layer convolutional network whose input is unprocessed low-resolution images. The robot showed good aptitudes to detect obstacles and navigate around them in real time at speeds of 2 m/s.
Many years later, with the rise of GPU-computing capacities for efficient learning for CNN, NVIDIA through their paper *"End to End Learning for Self-Driving Cars"* [34, 35] made popular end-to-end methods as part of the PilotNet architecture (cf. Figure 2.9). The proposed approach is to train a Convolutional Neural Network (CNN) to map raw pixels from a single front-facing camera directly to steering commands. The system learns to drive in traffic on roads with or without lane marking with only images from a front-facing camera coupled with the time-synchronized steering angle recorded from a human drive as training data. The authors stated that end-to-end system performs better then classical methods because the system internal components is self optimized to maximize the overall performance which is better then optimizing a selected module e.g., path planning, decision making, etc. The motivation for PilotNet was eliminating the process of hand-coding rules and allow the system to learn by observing. Similar end-to-end architectures have been reported in [26, 53, 204] and they differ mostly in the use of sensor inputs or problem space.
Most of the end-to-end methods in the literature predominantly utilize Deep Neural Network (DNN) to train off-line real world or synthetic data [26, 34, 53, 204] or Deep Reinforcement Learning (DRL) that are usually trained and tested in a simulation as the work of [117, 176]. Some other works try to make use of DRL for real world driving [125]. Pan et al., [169] tried to go one step closer to real test by first training a Generative Adversarial Network (GAN) to generate real-looking images from the synthetic images of the simulation and then give these generated images as input to the RL algorithm. Techniques

for porting trained DRL models from simulation to real world driving have been proposed in [125]. Imitation learning have been also widely used for end-to-end learning. In [170], a model predictive controller is used as an expert to generate optimal trajectory examples and exploited to train a CNN. Waymo through their Recurrent Neural Network (RNN) named ChauffeurNet [25] proposed exposing the learner to synthesized data in the form of perturbations to the expert's driving. The authors argued that standard behavior imitation is not sufficient for handling complex scenarios. Adding these perturbations have allow to create complex situations such as collisions and lead the learned model to be robust and able to drive a real vehicle in such critic situations.
In [56], the authors state that a vehicle trained end-to-end to imitate an expert cannot be guided to take a specific turn at an intersection which is why they proposed to use condition imitation learning on high level command input to output steering and acceleration.
In [230] introduced an approach to learning a generic driving model from large scan video data set. The model used a Fully Convolutional Network - Long Short Term Memory (FCN-LSTM) architecture to learn from driving behaviors. The driving model is evaluated based on the continuous/discrete feasible action prediction across diverse conditions. In the same manner to the previous cited research, the work given in [96] proposed to learn a driving model using a route planner (OpenStreeMap and planned route on TomTom Go Mobile) and a surrounding view of the vehicle with a 360-degree camera input as they argue that human drivers also use rear and side views mirrors when driving. For deeper analysis and review, the reader may refer to the extensive work and various surveys that exist in the literature concerning end-to-end learning such as [88, 130].

Although the idea of Dean Pomerleau [177] for end-to-end driving is very impressive, the need for guaranteeing functional safety is essential in self-driving car, something that AI has difficulty with. This is mainly due to the black box problem that doesn't have the transparency of their model-based counterparts. Safety of Deep Learning in Autonomous Driving will be discussed in Chapter 3.

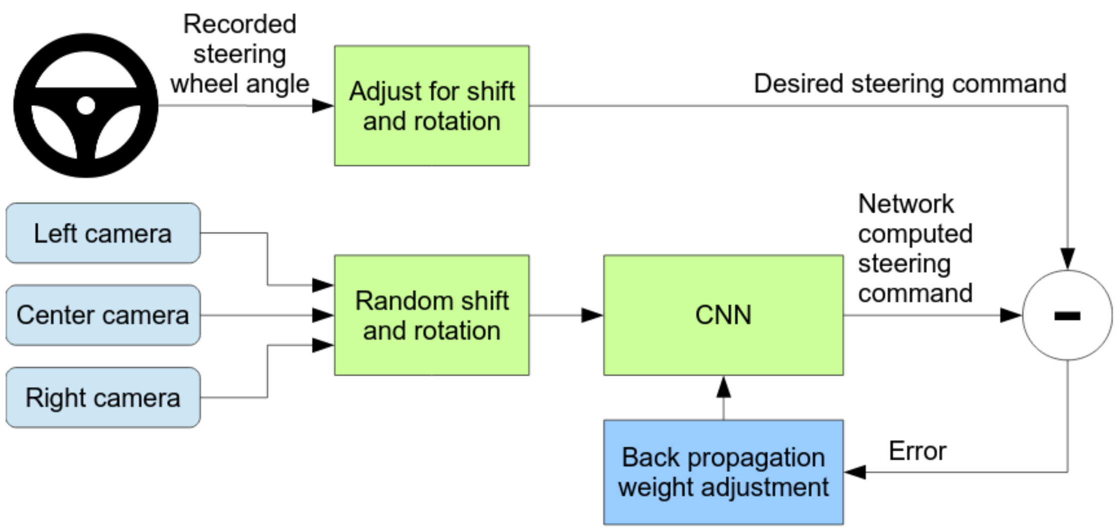


Figure 2.9: System architecture of PilotNet (Image credit [34]).

2.5/ Hybrid architectures: Combination of classical concepts and AI-based approaches

According to Judea Pearl [172], current AI systems only operates in a model-free mode which entails severe theoretical limits on performances as he states that such systems cannot have a retrospection reasoning and cannot thus serve strong basis for AI. For this reason, because the purpose in the conception of an intelligent system consists in trying to imitate the inference process of humans, model-free learners need the guidance of a model of reality. He proposed thus to equip machine learning systems with causal modeling tools through graphical representation that have made model-driving reasoning computationally possible, and thus represent a good basis for strong AI.
Model-Based Machine Learning (MBML) [33] defines this combination. Typically in MBML, a model is built automatically based on a set of assumptions that are written using a specific representation (e.g., graphical). These assumptions represent the variables in the problem domain, that affect each other. Thus, to go from the model to the predictions of these variables we need to account for the data and compute those variables values. This process is know as inference. Among these techniques of inference there is Bayesian inference. Bayesian Networks (BNs), falls under the MBML definition because they are considered as a probabilistic graphical language suitable for inducing models from data aiming at knowledge representations and probabilistic reasoning under uncertainty [107]. In [145], the authors state that BNs possess the property of being both a machine learning knowledge-based representation and a model-based formalism. Indeed, it allows structuring domain knowledge while accounting for dependencies between variables. This is also why many works classify them as knowledge-based approach [111, 185].
BNs have been successfully applied to solve a variety of problems in many different domains mainly related to modeling and decision-making under uncertainty [163]. In this Ph.D. work, we are interested in the use of BNs in order to be applied to AVs. Unlike other AI approaches like neural networks that needs extensive amount of data and learning time, the BNs have short response time given their computational tractability (for relatively small networks) due to the exploitation of conditional independence relationships [163, 186]. On the other hand, finding a realistic mathematical model that is able to understand the environment and its dynamic and make real-time decisions is not a simple task. BNs are also able to handle uncertainty that may arise from uncertain observations or the situational model. In contrast, neural networks can solve problems with uncertainty however massive data should be available.

Schubert [186] uses a Bayesian network for lane change decision-making and a Deceleration to Safety Time (DST) as a threat measure to assess the danger of the navigation lanes status. The proposed system perceive the vehicle's environment by processing images for lane and vehicle detection, estimate the state of the ego and the surrounding vehicles and take maneuvers decisions under uncertainty.
In real world application, data is limited in quantity and is usually gathered for a specific task or scenario. Path planning, trajectory prediction and control demands real-time performance, which means for deep learning usage, the time consuming data collection and training procedure should be simplified for online systems. Even with these conditions, proof of stability of the control system is necessary and how to generalize the deep learning models to all cases is still a challenge [199] as a single false decision can lead to a critical situation. Model-based approach in the other side and can guarantee the

satisfaction of some metrics/criteria.

As stated in Section 2.2, two main paradigms exist to address autonomous driving systems: *mediated perception* and *behavior reflex*. A third paradigm have been proposed in [52] that is called *direct perception* approach and falls in between the two previous paradigm. Their main idea (cf. Figure 2.10) is to map an input image to a small number of key perception indicators that directly refer to the accessibility of the road/traffic state for driving such as the angle of the car relative to the road or the distance to the lane marking. It is based on a deep Convolutional Neural Network (ConvNet) framework to automatically learn image features and estimate 13 indicators for driving. Based on these indicators and the speed of the car, a controller computes the driving commands to autonomously drive the car in different tracks of TORCS an open racing car simulator [228]. However, this approach was developed within a simple driving simulator and it would probably not generalize well when applied to much more complex real-world scenarios.

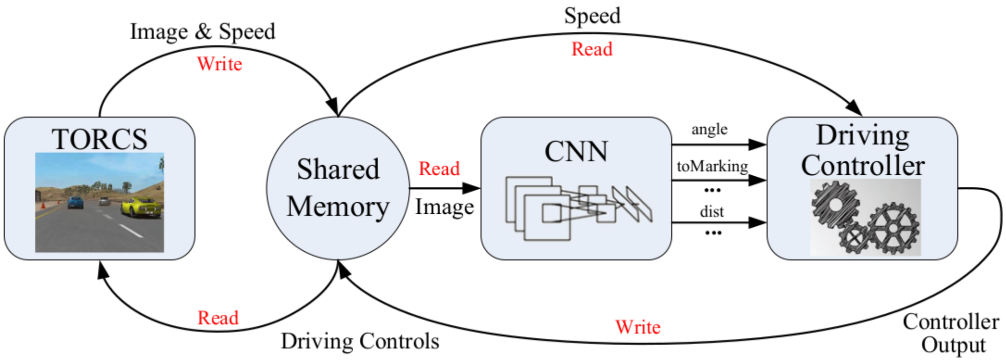


Figure 2.10: System architecture of a direct perception approach by Chen et al., (Image credit [52]).

Caltagirone et al in [48], generated driving paths by integrating lidar point clouds, GPS–inertial measurement unit (IMU) information, and Google driving directions. The system is based on a fully convolutional neural network that jointly learns to carry out perception and path generation from real-world driving sequences. This is accomplished by implicitly learning from theses driving sequences which areas of a point cloud are drivable and how a driver would navigate through them. In comparison with end-to-end approaches, the cited approach preserves the decoupling between perception and control.

In [158], it is proposed to combine the benefits of a classic driving system architecture with an end-to-end driving approach. Indeed, simulation in end-to-end driving systems brings up the problem of transferring the driving policies to the real-world. The key idea to resolve this issue is to encapsulate the driving policy such that it is not exposed directly to raw perception input. The architecture shown in Figure 2.11 is organized into three major stages: perception, driving policy that maps from a semantic segmentation to a local trajectory plan specified by waypoints that the car should follow, and low-level control. The system is trained in simulation using CARLA simulator [66] by training a driving policy from a per-pixel semantic segmentation of the scene to output high-level control. The trained driving policy is then transferred to a robotic trick in a variety of conditions. However, the shown results were applied to a simplified scenario and further extension is needed to make it applicable to real autonomous vehicles.

Some other works try to make use of what is called deep learning model (e.g., deep

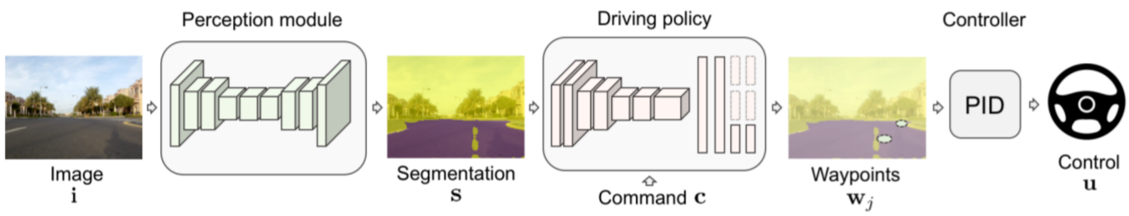


Figure 2.11: System architecture composed of a perception module implemented by an encoder-decoder network, a command driving policy implemented by a branched convolutional network and a low-level PID controller (Image credit [158]).

model predictive control [199] [22] or model-based Reinforcement Learning [89]). For instance, this method states that interactions of the system with the environment could be used to learn policies, value functions or even a model. However, learning a model introduces extra complexities and can induce model errors.

2.6/ Conclusion

In the light of the investigated literature, Table 2.1 illustrates a comparison between performances of the classical, the end-to-end and the hybrid approaches. Further, this table summarized what has been already said in each part through this chapter. Aspects considered in the depicted comparison are the most important requirements in today's AV.

Approach	Algorithm Name	Problem Space	Safety	Generalization and formal verification	Low Computational complexity	Handling Uncertainty	Experiments
Classical approaches	Tartan Racing Team DARPA 2007 [210]	Autonomous driving in urban environment	+	++	-	+	Real & Simulated
	Bertha-Benz Memorial Route in 2014 [236]	Autonomous driving in urban environment	++	++	+	+	Real & Simulated
End-to-end approaches	PilotNet by NVIDIA [34]	Autonomous driving in real traffic situations	+	- -	-	+	Real & Simulated
	ChauffeurNet by Waymo [25]	Autonomous Driving in multiple urban configurations	+	- -	-	+	Real & Simulated
Hybrid approaches	Bayesian Networks [186]	Decision-Making for autonomous driving in Highway	+	+	++	++	Simulation and Real experiment of the perceptive and the decision making modules
	The direct perception approach [52]	Autonomous Driving in multiple configurations	++	-	+	+	Testing on real-world data and on TORCS simulator [228]

Table 2.1: Comparison between model-based and AI approaches for AVs

A classification of the AV problem space is proposed in connection with the mentioned requirements. For each category two examples of the most pioneering algorithm are shown.

Accordingly, classical approaches have power in their ability to generalize and formally verify the overall algorithm giving that the algorithm are analytically defined. Also the presented architectures are very modular. End-to-end approaches on the other side do not have the ability of modularity or generalization as it is very dependent on the data used for learning but it has the strength to be compact and self-optimized as all the modules composing the classical architecture are integrated. The hybrid approaches have the utility of dealing with the limitations raised from each one of the methods. If the good equilibrium between both approaches is found, we believe that this can be the basis of a powerful system.

In the next chapters, safety assessment and decision-making of AVs which constitute among the main components of an architecture will be discussed. We aim to clarify the goals of this work, by discussing the most pioneering solution found on today's research.

3

Safety Assessment and Management of AVs

Contents

This chapter details the state of the art and bibliographical review on the risk management strategies for AVs, one of the major building blocks composting control architectures (cf. Chapter 2). Risk estimation and safety verification techniques will be presented and a classification is proposed of the different existing approaches.

3.1/ Introduction

The notion of risk is not always a clearly defined concept. The term "Risk" is used in many domains and is given different definitions depending on the field and context. Indeed, the notion of risk is of central concern to the theory of human decision making under uncertainty. Over the last two decades, the notion of risk and risk assessment have been thoroughly investigated. These efforts have resulted in an axiomatic theory of risk in finance and operations research [71]. Further theorizing of this understanding, leads to a widespread acceptance of coherent risk metrics that are used in many finance market. Parallel to the efforts in the finance community, risk metrics have been used in several other fields. We have been interested in investigating risk in robotics and the domain of autonomous vehicles. Majmudar et al., in [147] address the issue on how should a robot quantify risk and what constitutes a "good" risk metric. The authors come to the conclusion that a risk measure is said to be coherent if it satisfies six mathematical axioms that where highly inspired from the finance community. Nonetheless, this study theorize the risk metric assessment. However, when it comes to the physical meaning, theses axioms do not offer an intuitive understating for risk metric and ultimately lead to a difficult parameter setting [133]. Indeed, for many applications of mobile and manipulative robotics, the risk is generally associated to the probability of collision. In order to be safe, the robot has not to collide with other robots [73]. Of course, the risk is also very dependent of the application. For example in neurosurgery, since the brain consists in a number of regions and functionality, risk level differs essentially on which zones of the brain the surgical tool goes through [47].

Coming back to our field, while it may not be difficult for human drivers to tell whether a situation is safe, it is far from obvious for an autonomous car [91]. In that context, it is associated with the idea of whether the situation is or will be dangerous for the driver and for other traffic participants. Thus, it is natural to consider collisions as the main source of risk, and to base the assessment of risk solely on collision prediction [139]. Therefore a maneuver is said to be safe if no potential collision is possible and risk could be intuitively understood as the likelihood and severity of the damage that a vehicle of interest may suffer in the future. From this understanding, to assess risk it is necessary to predict how a particular situation will evolve. This is performed while using motion models [121, 129, 194]. To summarize, the widespread approach to collision risk estimation in the Intelligent Transportation System (ITS) community nowadays is composed of two steps:

1. Predict the potential future trajectories for all the moving entities in the scene while using motion model.
2. Detect potential collisions between pairs of entities and their predicted trajectories, and derive a risk estimate based on the overall possibility or proximity of the collision and the aforementioned state prediction.

This is called the situation or risk assessment layer in a classical control architecture (cf. Figure 2.1) and follows generally the decision making layer where based on the risk estimate of the previous steps, decide what is the most suitable maneuver.
In this chapter, we review the state of the art of risk assessment and safety verification techniques in the domain of Autonomous Vehicles (AVs).

3.2/ Motion modeling and prediction for AV's risk assessment

In order to make safe driving decisions, an autonomous vehicle must be able to accurately predict the changing traffic situation in order to asses the risk of driving. This problem is commonly known as motion prediction. It is used to infer the intentions of the surrounding drivers and predict what their state will be in the future timesteps.

This domain has been the center of interest of numerous works in robotics. An important number of works exists that reviews existing approaches from a mathematical point of view. In this chapter we will relate to the classification proposed by Lefevre et al., [139] in a comprehensive survey that classified existing motion prediction approaches used for risk assessment of AVs into three distinct parts: Physics-based motion model, Maneuver-based motion model and Interaction-aware motion model. The aforementioned classification will be updated with recent work in the domain.
The physical entities are the simplest and they represent vehicles as dynamic objects governed by the laws of physics, the maneuver-based entities on the other hand consider the future motion of vehicles and relies on estimating the intentions of the other traffic participants based on either prototype trajectories or maneuver estimation and execution. For the two first categories, the assumption made is that all the traffic scene participants behave independently from each other, theses approaches lack contextual meaning as the surrounding environment is not taken into account. In contrast, interaction-aware model exploit the inter-vehicular relationship by incorporating contextual information when modeling the motion of the vehicles in the scene. As pointed out in [121], the vast majority of the interaction-aware model are build using Bayesian Network (BN) due to their ability to model relationships between variables.
In what follows a summary for each of the aforementioned categories will be provided with some explanatory examples.

3.2.1/ PHYSICS-BASED MOTION MODELS

Trajectories are predicted by propagating the initial vehicle state over time based on dynamic and kinematic models linking the evolution of the sate of the vehicle known as the pose (e.g., position and heading) with: the control output (e.g., acceleration, speed, steering), external condition (e.g., friction coefficient), car properties and noisy sensory data.
Extensive work and many surveys [139, 187, 194] exists on such motion models (also called evolution models) and they usually differs in the representation of the dynamic and kinematic of a vehicle and on how uncertainties are handled.
Dynamic motion models based on Lagrange's equations uses the internal parameters of the vehicles such as the forces acting on the wheels or the chassis to act on the vehicle's motion [178, 193]. In contrast, kinematic motion models is based on the mathematical relationship between the parameters of movement such as the position, velocity or acceleration. Common kinematic motion models include Constant Velocity (CV), Constant Acceleration (CA) for the simplest and Constant Turn Rate and Velocity (CTRV), Constant Turn Rate and Acceleration (CTRA) also known as Constant Yaw Rate and Acceleration (CYRA) for more complex models. A comparative study between different models is performed in [187]. These motion models are often chosen for trajectory prediction than dynamic motion models because of their simplicity and efficiency. Indeed, for this type of application, the parameters needed for dynamic models are not observable by exteroceptive sensors which exclude the use of dynamic models to predict the trajectories of other traffic participants.
In the literature, many ways exist to predict a trajectory and handle the uncertainty while using the evolution models. The easiest way to predict a trajectory is to apply an evolution model directly to the current state of the vehicle while assuming a perfect knowledge of this state and perfect representation of the vehicle through the evolution model. This kind of prediction doesn't account for uncertainty on the current state which makes it not

reliable for long term prediction. However its computational efficiency makes it suitable for real time application.
The other way to handle the uncertainties corresponds to the the Gaussian noise representation through the use of the well-know Kalman Filter. In this case the state of the vehicle is recursively estimated from noisy sensor measurements and the uncertainty on the current vehicle state and on its evolution is modeled by a normal distribution. The predicted trajectory is gathered by looping on the prediction step of a Kalman filter during a pre-defined prediction horizon in case that the driver maintains the same driving profile [18]. This last point represents also a limitation concerning this type of prediction algorithm because this is insufficient to represent different possible maneuvers. Some works use a Switching Kalman Filter (SKF) [160], also know as Interaction Multiple Model (IMM), to palliate this limitation. It used a bank of Kalman filters components, each one adapted to a specific behavior (lane keeping, lane changing, intersections) and uses a group of motion models that capture the dynamics of the targeted behavior.
Monte Carlo simulation [15] is an alternative technique to model the uncertainty in the predictions. It is particularly useful when no assumption is made about the linearity of the model or the type of the statistical distribution of uncertainties. In summary, most methods sample from the input variables of the motion model instead of considering them to be constant to generate a possible future trajectories. The Physics-based motion models are the simplest to use, however they are not able to reason on a high-level basis about a situation and therefore are limited to short-term collision prediction.

A summary of what have been exposed in this sub-section is given in Table 3.1.

3.2.2/ Maneuver-based motion models

To palliate to the short-term abilities of the previously mentioned motion models, the maneuver-based entity chose do adapt the predicted trajectory to the on going maneuver over a longer prediction horizon. Multiple work in the literature [104, 194, 238], include a maneuver recognition function upstream of the trajectory prediction. The goal is to recognize a particular maneuver from the current state of the vehicle and possibly from the object's past states. The methodologies used are various, they go from heuristics [104] to learning algorithms such as neural network [140] passing by logistic regression or support vector machine [149]. One of the most popular method for this kind of use is Hidden Markov Model (HMM).
In [135], a set of four driving maneuvers is defined: going straight, turning left, turning right and passing. For each of them, a trajectory is also predefined. Then, from these trajectories and the data sequence formed by the last measurements carried out, the occurrence response of each of the predefined maneuvers is estimated, using a HMM. The sensor measurements contain parameters such as the distances of the vehicle from the left and right limits of the lane, the activation of the turn signals, its speed or its proximity to an intersection. The trajectory is defined by a sequence of similar parameters. The HMM algorithm assigns an output probability at each predefined maneuver, from the current measurement sequence. The recognized maneuver is therefore the one with the highest probability.
In [104], the authors propose a Maneuver Recognition Module (MRM) based on the comparison of the center-line of the road's lanes to a local curvilinear model of the path of the vehicle. Mahalanobis distance is then used to compare the properties of each trajectory and to select the most likely maneuver from a predefined set.

Various methods are also used for the maneuver execution part and prediction. Trajectories are predicted in order that they match the identified maneuver. For example the previous cited work [104] uses for the trajectory prediction, a combination of a trajectory prediction based on Constant Yaw Rate and Acceleration (CYRA) motion model and a trajectory prediction based on maneuver recognition. The MRM predicted trajectory is selected from a set of generated trajectories with respect to a cost function. The set is built based on the vehicle current state, the road parameters and the detected maneuver. Other work uses Rapidly-exploring Random Tree (RRT) [20], Stochastic reachable sets [17], Gaussian Process [135] or Long Short-Term Memory (LSTM) neural network [13, 127] which are often used for this purpose. The readers may refer to [139, 180, 226] for more details about these predictive algorithms.
Even though, maneuver-based motion models solves the issue of the horizon of prediction making it longer but remains considering the interactions between vehicles to be compatible with the way human driver makes driving decision where transitions between vehicles states are highly coupled.

3.2.3/ Interaction-aware motion models

The need for modeling interactions of different behaviors of the vehicles arise among the first in the work of Dagli et al,.[60] where instead of relying only on the system dynamics, they proposed to infer the motivations and goals of driver in order to predict his behavior. The literature is various in this domains and many approaches have been proposed to include the estimation of the behaviors of other traffic participants and their interactions [150, 152, 222, 234]. However, we will focus in this section in discussing modeling interactions with Dynamic Bayesian networks (DBN) as it consists in the wider used methodology for modeling interactions between traffic participants in the AVs literature.

DBN [160] are very used for this kind of motion models [122]. DBNs are an extension of Bayesian Networks (i.e., a graphical representation of a joint probability distribution of random variables, cf. Appendix A) to handle temporal sequential data and interaction between variables. HMM [152] and Interacting Multiple Models (IMM) filters [150] can be seen as particular case of a DBN.

Gindele et al., [82] suggested to use a DBN for predicting traffic participants' behaviors and trajectories in highway scenes. This is achieved by recognizing the type of situation derived from the local situational context that describes the properties of the vehicle surroundings and subsumes all information relevant for the drivers decision making. The resulting DBN is shown in Figure 3.1. Continuous nodes are depicted by circles and discrete nodes by rectangles. Solid lines represent direct dependencies within a time frame while dashed lines state dependencies between time steps. X is the the state of all traffic participants that is estimated using Z the measurement vector. On the other side C, the vector containing a set of defined features relative to the actual situation context, e.g., the distance to next vehicle ahead, is modeled as a function of X. The set of behaviors B depends on the current situations S and the last conducted behaviors. From the behaviors, the trajectories realization T are produced. Inference, in this probabilistic model is performed using particle filtering.

The work done by Lefèvre et al., [138] aimed at including a motion model that takes into account traffic rules and inter-vehicles interactions along with collision risk estimation. This paper explored the strength of interaction-aware models by implementing a DBN

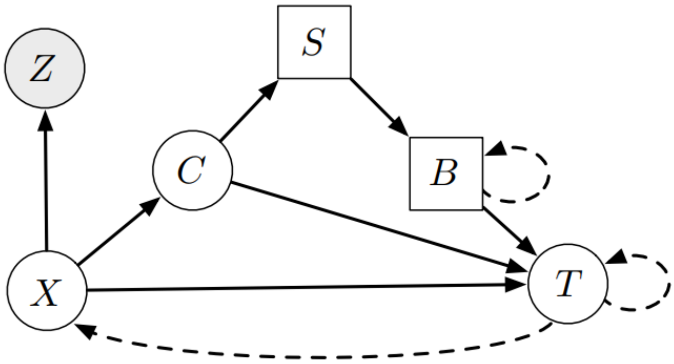


Figure 3.1: DBN proposed by Gindele et al., (Image credit: [82]).

for risk assessment at road intersections. Instead of predicting all surrounding vehicles' trajectories, only vehicles that were found to disobey traffic rules were evaluated for possible collisions. They are identified by detecting conflicts between what drivers intend to do and what is expected from the. It was assumed that Vehicle-to-Vehicle (V2V) communications were allowed so that vehicles could communicate their poses, speeds and steering movements by the means of wireless communication links. However, communications between vehicles is not always guarantee and the developed models require a lot of parameters to be tuned. Even with these requirements, current approaches cannot cope with traffic scenarios of high complexity.

By making use of the interaction aware formalism, Schulz et al., [189] proposed a multiple model Unscented Kalman filter (UKF) based DBN inference method for driver intention estimation and multi-agent trajectory prediction. A probabilistic forward simulation of the network's estimated belief state is performed to generate the different scene developments. This provides the corresponding trajectories for the set of possible future scenes.

The interaction-aware methods based on DBNs constitute the most advanced motion prediction technique in the literature due to their flexibility to model interactions and respect contextual feature like road network. However inference on these models is in general computationally expensive.

After predicting the potential future trajectories of all the moving entities the next step is to effectively detect collision between pairs of entities and their predicted trajectories. Many methods have been proposed in the literature and we propose in what follows, a classification of the risk assessment methods.

3.3/ Collision-based risk assessment

The risk of a situation can be estimated by analyzing the actual driving configuration in the environment, foresee probable changes in the action of other vehicles and predict potential motion of the vehicles. Given these predictions, extracting information on the possible occurrence of a collision is possible. While the simplest techniques provide basic methods on whether and when a collision will occur, more complex methods can compute in addition an information on its probability or its severity. In this section we divided the risk metrics in two categories the deterministic risk indicator and probabilistic collision prediction.

Approach	**Methods**	**Algorithm and tools**	**Advantages and limitations**
Physics-based motion model	Kinematic and Dynamic models	Kalman Filtering, Monte-Carlo simulation	+ Simplicity and Computational efficiency - Short-term prediction
Maneuver-based motion model	Maneuver Intention estimation coupled with trajectory prediction	Hidden Markov Model (HMM), Heuristics [104], Classification of motion patterns [139, 141], Support Vector Machines (SVM) [149]	+ Longer prediction horizon - Inter-Vehicles interaction disregarded
Interaction-aware motion model	Trajectory prediction by including inter-vehicles interaction	Dynamic Bayesian Network (DBN)	- Computationally expensive + Consider interaction between road participants + Design Flexibility

Table 3.1: Motion Models and Prediction: A summary (adapted from [139])

3.3.1/ Deterministic risk indicators

This first approach uses simplified models of evolution (cf. Section 3.2.1) in order to compute risk metrics. The most-known indicators of criticality are: the change in velocity of the vehicles, the collision angles (rear-end or head-on), the amount of overlap between different shapes representing vehicles (ellipses, circles , polygons, etc.) [81, 102, 173], the occupancy of conflicting areas [188], the rate of change steering, the configuration of trajectories in a collision course, the remaining time span in which the driver can still avoid a collision by braking (e.g., Time-to-Brake or by steering, etc.)
In the literature there exists several risk metrics [32, 38, 98, 124, 166, 184] that cover these indicators of criticality. Table 3.2 presents some of the most-known deterministic risk measures used in nowadays AVs. They are classified in a chronological way of appearance in the literature as well as in three categories which are: the longitudinal risk assessment, lateral risk assessment and both at the same time. A unique warning threshold is usually applied to the obtained value to classify it as risky or safe. Some other works [137, 220] distinguish multiple warning levels that improve the decision making process and conform to the driver perception of safety in a dynamic environment.

Despite the good number of metrics that exist in the literature, the TTC remains the most used and the oldest risk metric to the best our knowledge in the domain of AVs. This is mainly because of its simplicity, its low cost computational time and it has a lot of variants for many different configurations [136]. However, as pointed out by Laugier et al., in [135] this metric suffers from the lack of context and singularities may occur in some configuration. Indeed, TTC alone is insufficient as a risk indicator for managing complex situations. In addition, the common definition of the TTC is restricted for a specific path to detect longitudinal collision and for well-defined scenarios such as car following. To overcome this issue, extended definitions of the Time to collision have been proposed in multiple works [102, 118, 153, 223]. The work given in [1] and [102]

Approach	Risk measure	Description	Equation
Longitudinal Risk Assessment	Time-to-Collision (TTC) (Hayward 1972) [95]	The time required for two vehicles defined with their pose, velocity and wheelbase denoted as (X_i, V_i, L_i) to collide if they continue on at their present speed and path.	$TTC = \frac{X_1 - X_2 - L_1}{V_1 - V_2}$ if $V_2 > V_1$ for the case of rear-end collision (see [136] for other configurations).
	Post-Encroachment Time (PET) (Allen et al., 1977) [12]	The time between the moment t_1 that the the first vehicle last occupied a position and the moment t_2 that the second reaches the the same position.	$PET = t_2 - t_1$
	Deceleration to safety Time (DST) (Hupfer, 1996) [110]	The deceleration a_{DST} that has to be applied to the **e**go vehicle velocity v_e to maintain a certain safety time t_s with respect to the **o**bject vehicle velocity v_0.	$a_{DST} = \frac{3(v_e - v_0)^2}{2(x - v_0.t_s)}$ with x the distance between both vehicles. (see [186] for more details)
Lateral Risk Assessment	Time to Line Crossing (TLC) (Godthelp 1984 [84])	The time duration t_{LC} available for the driver before any lane boundary crossing.	For straight road configuration and zero steering angle: $t_{LC} = \frac{y_{ll}}{v_l}$ with v_l the lateral velocity and y_{ll} the lateral distance of front left tire to the line that would be crossed. (see [84, 148] for more configurations)
Longitudinal & Lateral Risk Assessment	Time To React (TTR) (Hillenbrand et al., 2006) [98]	The remaining time to avoid an imminent collision by emergency braking with full deceleration (which is known as Time to Brake (TTB)), steering with maximum lateral acceleration (Time-To-Steer (TTS)), or a kick-down maneuver (Time-To-Kickdown (TTK)) by leaving the collision zone early enough for example.	The TTR is calculated as the maximum of the TTB, TTS and TTK [98]

Table 3.2: Some examples of deterministic Risk measures

addresses the problem from a planar perspective where vehicles are considered in a two-dimensional plane and the state of each vehicle is defined by a vector position, velocity and acceleration components on X and Y direction. This Extended TTC (ETTC) is computed at each time step for each vehicle pair that are close enough.

Even though a deterministic approach has less computational complexity, the main drawback of these criteria is the incapability to deal with the unpredictable in short-lasting maneuvers, such as lane changes.

3.3.2/ Probabilistic methods for risk assessment

These approaches takes into account the uncertainty of motion along the predicted trajectory (cf. Section 3.2.3 and 3.2.2). Several probabilistic frameworks have been used: Hidden Markov Model (HMM) [70], Dynamic Bayesian Network (DBN) [142] or other probabilistic frameworks.

In Lefèvre et al work [138] as stated before an interaction aware formalism have been used for the trajectory prediction and the maneuver intention prediction used for the risk assessment. Dangerous situations are identified by detecting conflicts between intention and expectation, i.e., between what drivers intend to do and what is expected of them. It is formulated as a Bayesian inference problem where intention and expectation are estimated jointly for vehicles converging to the same intersection. Risk is computed as the probability that expectation and intention do not match.
In [121], it is proposed to integrate what is called in the paper a network-level collision prediction through a Random Forest classifier with interaction-aware motion models under a Bayesian framework DBN for risk assessment of AVs. The network-level collision prediction consist of the safety context (safe or collision-prone) of the road segment on which the ego-vehicle is traveling on and interprets a traffic scene as either “dangerous” or “safe” enabling a vehicle to be more vigilant in collision-prone situations.

Other research used the reachable set computation in order to asses collision in future planned path [17]. Reachable set is a set that contains all possible states that the system trajectories can evolve into. A collision is verified by checking whether the reachable positions of the ego vehicle intersect the reachable positions of another vehicle. A geometric version of reachable states has been proposed in [87], that measures the collision probability as the percentage of overlap between the geometric shapes representing the future motion of vehicles. However, with the classical definition of reachable set, other vehicles rapidly cover all positions the autonomous vehicle could possibly move to. Which results in the fact that the planned paths of autonomous vehicles are often evaluated as unsafe. For this reason, the reachable sets are enhanced by stochastic information that gives the probability of the crash and it is called *stochastic reachable sets* [16].

Broadhurst et al., in [42] chose to use a Monte Carlo path planning algorithms to generate a probability distribution for the future motions of traffic scene participants. Each object is assigned a goal function, and Monte Carlo integration is used to evaluate the probability that all the objects would safely achieved their goal.

The grid based approach is another way to assess the risk. It consists in constructing grid cell values from sensory information. It is used to model the environment and propose to split the space into a set of cells that may be free or occupied. Usual methods aim to calculate the probability of occupation of a cell from sensor data. It was first proposed by Elfes in [69]. Bayesian Inference is the common used methods to cope with uncertainty and errors. Many extension have been published in the literature. For example the Bayesian Occupancy Filer (BOF) used by [59] provides filtering, data fusion, and velocity estimation capabilities of the cells while allowing parallel computation.

Evidential grids [156, 168] are a variation of occupancy grids. It is is based on Dempster-Shafer (DS) theory and offers a solution to make the difference between unknown and doubt caused by conflicting information output from the fusion process.

3.4/ BEHAVIOR-BASED RISK ASSESSMENT

Collision based risk indicators defined previously relies on a rather unconstrained evolution of the environment. It estimate the risk of a situation by predicting the future trajectories mostly based on the current state and then look for a collision between these trajectories.
In this section we will discuss some interesting approaches named Behavior-based risk assessment, that estimate the risk of vehicles deviating from their nominal behavior expected on the road.

Hundreds of research has been done in the domain of warning and detection systems [51, 195] of incoherent or unexpected events done by vehicles in the environments mostly for ADAS. When dealing with AVs, the behavior-based risk can mostly be estimated by either:

- Defining a nominal behavior of vehicles and then detect event that doesn't match,
- Detecting conflicting intentions between vehicles or with regards to traffic rules,
- or by dealing with unexpected behavior as independent events while using safety verification techniques upstream of the risk assessment modules for emergency situations which will be discussed in Section 3.7.

The behavior-based risk depends heavily on whether all the vehicles in the environment are self-driving car or not. In this section we will concentrate our reviewing efforts on the methods used when dealing with human drivers to assess the risk.

In [182], the authors proposed to model the activity at vehicular intersections with the aim of detecting unusual events such as red-light infringements and forbidden turns. A method is build for detecting, tracking, and modeling the behavior of moving objects while using the specific constraints that this kind of scenarios have. It is based on a dual layer. The first one deals with appearance at the intensity level. The second one with the different moving directions present on an image sequence. During operation, a particular observation is compared with the learned model of a normal behavior of vehicles in the crossroads.
Lefèvre et all., in [138] proposed to identify dangerous situations by comparing what drivers intend to do with what they are expected to do. Driver intentions and expectations are estimated from the joint motion of the vehicles, taking into account the layout of the intersection and the traffic rules at this intersection. This method can be categorized in both behavior-based and probabilistic risk assessment as the risk is computed as the probability that expectation and intention do not match. This kind of methods allows to avoid the extensive complexity of generating all the possible trajectories to detect collisions between each possible pair that is often used in collision-based risk assessment.

By making use of the interaction aware formalism, Schulz et al., [188] proposed a behavior prediction framework through a DBN, which explicitly considers the intentions of drivers and the inter-dependencies between their future trajectories. The decision making process of an agent is divided into three hierarchical layers: which route it is going to follow (route intention), whether it is going to pass a conflict area at an intersection before or after another agent (maneuver intention), and what continuous action it is going to execute. Unexpected behaviors are not included explicitly in the framework however the inter-dependencies between all the vehicles future trajectories are taking into account which allows to deal with conflicting areas and act accordingly.

Other work, dealt with mechanical failure management that cause vehicles to deviate from their trajectories. On way of doing is to prevent vehicles from following their course after the failure, however, this approach cannot prevent collisions among vehicles that are already in dangerous configuration like intersection or during lane changes. In [23] the authors proposed a method to account for several common types of mechanical failures at intersection by pre-computing evasion plans before vehicles enter an intersection. They argue that this approach is necessary because there are situations in which vehicles cannot evade without pre-allocation of space for evasion.

3.5/ OTHER METHODS FOR RISK ASSESSMENT

Other methods estimate the risk jointly within the path planning through algorithms (for example while using optimization approaches) based on a chosen trajectory considered as safe with respect to certain constraints related to the vehicles dynamic, the road geometry, the dimension of the vehicle or the occupancy of objects in the environment. In this kind of application one can make the analysis concerning the constraints defined in the optimization and the used algorithm.

On-board of the vehicle that completed the 103 km of the Bertha-Benz-Memorial-Route fully autonomously, Ziegler et al., [235] proposed to use an optimal trajectory generation based on a continuous optimization has been used. The solution trajectory is the constrained extremum of an objective function that is designed based on the dynamic feasibility and comfort. Static and dynamic obstacle constraints are incorporated in the optimization in a form of polygons. The constraints are designed in such away that the solution converges to a single global optimum.

Pek et al., in [173] developed a fail-safe trajectory planner for self-driving vehicles. This trajectories are computed in real-time in continuous space by making use of convex optimization techniques. This allows to separate motions into a longitudinal and a lateral component while defining the constraints suitable for each motion and thus guaranteeing the drivability of the resulting motions. Collision avoidance is done through the convex constraint set that considers the kinematic vehicle model, which is described with respect to a curvilinear coordinate system aligned to a reference path and restricts solutions so that they do not intersect with the predicted occupancy sets of other traffic participants.

Some other methods chose to use a hybrid approach by combining multiple approaches. Kim et al., in [127] propose to use a recurrent neural network called Long Short-Rerm Memory (LSTM) to analyze the temporal behavior and predict the future coordinate of the surrounding vehicles from the coordinates obtained from sensor measurements. From the LSTM, the proposed scheme shown in Figure 3.2, produces then the probabilistic information on the future location of the vehicles over occupancy grid map.

Many other works combine the efficiency of a deterministic criteria like the TTC with an interaction-aware formalization while using a DBN. It is the case for exemple of the works done in [83, 121].

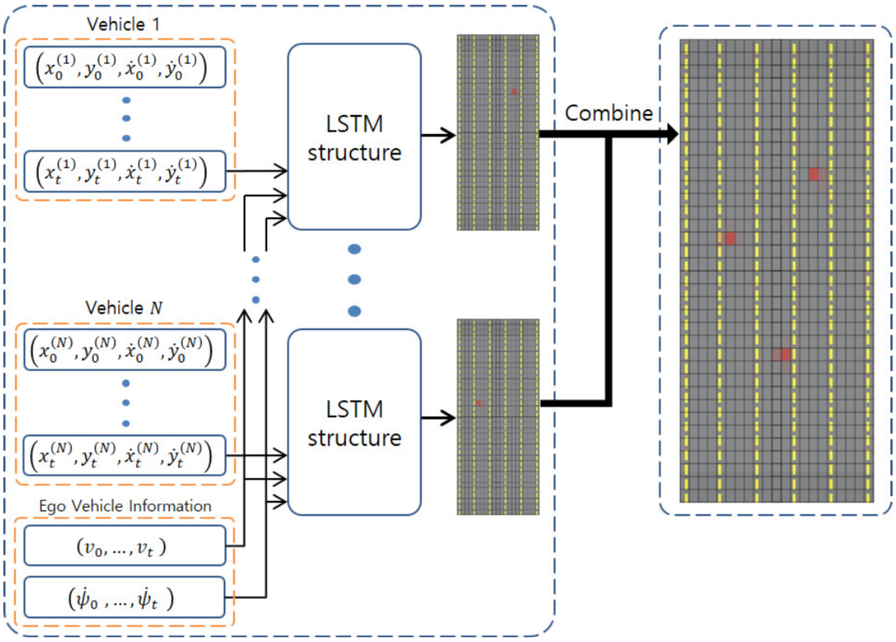


Figure 3.2: The structure of the proposed scheme (Image credit: [127]).

3.6/ Safety of deep learning approach for autonomous driving

Decision-making system in AI is either solely based on machine learning in an end-to-end fashion (cf. Section 2.4) or involves some combination model-based reasoning and machine learning components (cf. Section 2.5). Either way, the urge of explainable AI [90] able to guarantee safety is receiving increasing attention in nowadays research [19, 46, 108, 190, 213]. More and more AI and deep learning strategies are effective and reliable even for safety-critical related issues. Guaranteeing safety of a system running any kind of method or technique heavily rely on:

- the type of the used technique,
- the application context,
- the understanding of the impact of possible failures,
- the definition of what is a safe behavior,
- the definition of the assumptions and the constraints on the system and its environment,
- the uncertainty handling.

In the particular case of deep learning, in addition to the previous requirements the need of a justification explanation about the decision that has been taken and the insurance of safety on all possible driving situations remains subject of of highly active nowadays research.
Many work in the literature try to map this requirements for deep learning techniques by

defining a set of constraints and assumptions into the design of the used network that needs to hold in order to ensure the safety of the specified behavior. The work given in [46] for exemple, designed an application of CNNs to detect (i.e., classify and localize) objects based on camera images as part of a collision avoidance system for self driving vehicles. For example, standard specifications which could be obtained at the level of the machine learning feature could include: the class of the located object that are at a specified distance or with a certain lateral precision and that can be functions that depend on the velocity of the ego vehicle or TTC. These assumptions are mapped to the dimensions of the image frames and presented to the CNN.

On the other hand, some researches focus their effort in analyzing the faults and failures caused in a deep learning components. They may be due to unreliable or noisy sensor signals, neural network topology, learning algorithm, training set or unexpected changes in the environment. For example, Varshney in [213] reasoned about the definition of the term *"safety"* stating that it should be defined in terms of risk, epistemic uncertainty and the harm incurred by unwanted outcomes. Instead of reasoning about the input data, he analyzed the choice of cost function, the appropriateness of minimizing the empirical average training cost and the empirical risk and defined strategies to achieve safety.

In a same spirit, the work given in [19] attempted to analyze the problem from the point of view of *"accidents"* stating that these dangerous behaviors may arise from poor AI system design and proposed a list of five practical research problems related to accident risk, classified according to whether the problem results from having the wrong objective function (avoiding side effects and avoiding reward hacking), an objective function that is too expensive to evaluate frequently (scalable supervision), or undesirable behavior during the learning process (safe exploration and distributional shift).

A must cited example is the Tesla Autopilot accident in 2018, caused by a misclassification error despite the 130 million miles of testing and evaluation under extremely rare circumstance. Despite the fact fail-safe mechanisms exists [50], that must stop the autonomous vehicle in case a failure is detected, one of the main cause that lead vehicle governed by deep learning to crash are the mistakes done by lower-level components that propagated up to the decision making process and lead to devastating result. In the case of the Tesla, the low-level component failed to distinguish the white side of a turning trailer from a bright sky. In this kind of system, the uncertain information, in this case distinguishing between the sky and another vehicle, should be escalated to higher level decision and may advise the user to take control of steering. Gal in his Ph.D [77] worked on developing tools to obtain uncertainty estimation and confidence models in deep learning, casting recent deep learning tools as Bayesian models without changing either the models or the optimization. For the purpose of accounting for uncertainties, [151] proposed to use end-to-end Bayesian deep learning architecture for AVs. A comparison has been made between a framework built on traditional (non-Bayesian) and Bayesian deep learning. Although both systems use the same initial sensory information, propagating uncertainty through the prediction and decision layers allows the Bayesian approach to avert disaster.

In all these cases, the safety of a decision in deep learning can be reduced to ensuring the correct behavior of the system and many survey the techniques used and the uncertainty handling [77, 88]. Verifying the program is working as intended before deployment is possible, however, safety assurance and formal verification methodologies are still not clearly fixed yet.

3.7/ Beyond risk assessment: Guarantee of safety of AVs

3.7.1/ Safety verification of AVs

The common task for autonomous vehicles after the decision-making is to determine a nominal trajectory to perform lane changes and other maneuvers taking into consideration any constraints or traffic condition that are known at the time of planning. The procedure must therefore be aborted automatically in case of any unexpected approaching objects such as other objects and road users entering the planned course of the vehicle. The vehicle must then be able to replan by determining an alternate route, i.e., the emergency trajectory, which the car will pursue instantly to avert an accident and guarantee safety all the time. Extensive testing to simulate all possible behaviors of other traffic participants is a time consuming task. Indeed, considering the uniqueness of each traffic situation, the task of modeling every situation is nearly impossible. In addition, it can only prove that a system is unsafe, but is not able to propose an alternative. Classical safety verification techniques perform the safety verification offline before the vehicle is deployed and can only investigate a certain class of situations, as it is done for the verification of an automatic cruise controller in [197]. These techniques are usually called *scenario-based verification*.

Since every traffic situation is unique, it is necessary that the decided/planned maneuvers be always verified during navigation of the vehicle. This has been called in the literature online safety verification [14] or formal verification and answers to this challenge. It has been used in many works of the literature [14, 155]. In [14] reachability analysis is used as a safety verification method. The verification is carried out by estimating all potential occupancies set of the automated vehicle and other traffic participants. To capture all potential future possibilities, reachability analysis is applied to account for all potential behaviors of mathematical models including uncertain inputs (e.g., sensor noise, disturbances) and partially unknown initial states. Possible future collisions are identified when comparing the intersection of the obtained sets. However if the trajectory is regarded as unsafe no alternative is proposed to avoid the collision.

Spatial Logic reasoning has also been applied to safety verification, for proving collision freedom of lane-change maneuvers [99] or for real-time spatial logic that can specify constraints in traffic maneuvers on multi-lane motorway [229], however, modeling a generic model using logical formulas is a big challenge.

Pek et al., in [175] presented an approach for verifying the safety of lane change maneuvers of autonomous vehicles by incorporating formalized traffic rules. The assessment is based on safe distances, allowing the ego vehicle to drive safely for an infinite time horizon which allows the autonomous vehicle to verify its decision to change lanes and recover if the lane change becomes unsafe during the maneuver.

Safety Verification of Deep Neural Networks on the other hand, have been little studied and few works exists in this topic. Huang, et al., [108] proposed a framework for automated verification for the safety of classification decisions, which is based on search for an adversarial misclassification within a given region. The key distinctive features of this framework compared to existing work is the guarantee that a misclassification is found if it exists.

In a global manner, deep learning techniques have become increasingly popular in the domain of decision-making and autonomous navigation, however still remains questions about their ability to guarantee safety since their output response is not well known, particularly outside the training data scope. Authors in [190] identified some challenges about the verification process in artificial intelligence systems and they mainly consist in modeling issues.

3.7.2/ Evasive maneuvering in emergency situations

Because maneuvers are verified online while using safety verification techniques, the ability of the system to re-plan and evade a dangerous situation becomes possible. Emergency scenarios may necessitate maneuvering up to the vehicle's handling limits in order to avoid collisions [76].

The common used methods and the one from very early work related to emergency situations is to simultaneously plan a nominal and an emergency trajectory in order to guarantee the safety of the vehicle controller. With the help of this planning process the vehicle controller is able to provide an emergency trajectory before and during the performance of a lane change or any other maneuver. Several literature research tackle this problematic. Among the earliest of these [198] presents an emergency lane change maneuver as they state this maneuver is one of the many appropriate responses to an emergency situation. It is applied on a platoon of two vehicles. Hirsch et al., in [100] proposed a path planning method for emergency maneuvers of autonomous vehicles. A pre-defined nominal trajectory is analyzed with reference to obstacles modeled from circular safety areas. Based on the results of this analysis an emergency trajectory is calculated by using the method of elastic bands. In this paper a simple example is illustrated: a moving car approaches from behind, whose vehicle controller has calculated a nominal trajectory. However the detection of the obstacle initiates a calculation of the evasion trajectory which the moving vehicle traces immediately.
Magdici et al., [146] try to make use of what is called a fail-safe motion planner for autonomous vehicles which combines an optimal trajectory with an emergency maneuver. The authors proposed a module that generates an optimal path considering the most likely movement of leading vehicles for a given time horizon and for each time step an emergency maneuver is kept available which account for every possible maneuver of surrounding traffic participants. If there is no other further safe trajectory available, the emergency maneuver is applied, which can safely bring the ego-vehicle to a stop while avoiding any collision. However, generating an emergency maneuver for each time step is computationally expensive and often not needed.
In [5] the authors proposed a combination of steering and braking to construct an evasive maneuver in the purpose of avoiding a collision with a vehicle driving in front of the ego vehicle even at the last moment. To allow a still later intervention an evasive maneuver (lateral action) and the braking maneuver are combined. Two control strategies are presented for a combined intervention. Both methods have the goal to make use of the friction potential of the tires as much as possible. The first control strategy uses the accelerations which occur during the maneuver, the second control is a tire-individual slip-control. Thus it is possible to significantly reduce the evasive length in the speed range between 70 and 160 km/h compared to the classical evasive maneuver. With such a system, collisions can be prevented to an even later stage. In [24] a Nonlinear Model Predictive Control (NMPC) for emergency collisions avoidance in complex situations between a driver's vehicle and neighboring vehicles is presented. NMPC is used to predict

the trajectories of all vehicles, and if a collision is detected in the predicted trajectories, the driver's vehicle attempts to avoid the collision through a left lane change, right lane change, or braking maneuver.
Recent research [58] propose an emergency lane change based on a Markov process decision strategy with enhanced vehicle handling performance (vehicle dynamics and control aspects). The output control of the system is the steering of the vehicle.
An even detailed procedure for evasive action handling is proposed in [173]. Pek et al., developed a fail-safe trajectory planner for self-driving vehicles. This trajectories are computed in real-time in continuous space by making use of convex optimization techniques. This approach simultaneously favors jerk minimization through the defined motion models and their corresponding constraints and incorporates the safety verification in the planner in such a way that trajectories are always verified as safe. However, this approach lacks real experimentation to prove its efficiency.

A step forward into emergency situation management, is considering cases where the collision in unavoidable. Fraichard [73, 74] have worked on the subject by proposing a concept called Inevitable Collision State (ICS). An ICS is characterized as a state for which, no matter what the future trajectory followed by the system is, a collision with an obstacle eventually occurs. It takes into consideration both the dynamics of the system and the obstacles. The concept is useful both for navigation and motion planning purposes as for its own safety, a robotic system should never find itself in an inevitable collision state however determining ICSs is computationally intense and suffer from uncertain future motion of obstacles.

To palliate to this issue, authors in [174] introduce invariably safe sets which are regions that allow vehicles to remain safe for an infinite time horizon. A tight under-approximation of the proposed sets is obtained in real-time with respect to the number of traffic participants while maintaining formal safety guarantees. Moreover, these sets have been used to determine the existence of feasible evasive maneuvers and the criticality of scenarios by computing the time-to-react metric. The drawback around set-based methods is that they tend to be overly conservative. Thus, in dense traffic and a highly uncertain environment, these methods are likely to yield no suitable solution.

3.7.3/ Standardization and generalization of safety frameworks

In order to be compared, decision-making framework must be tested on the same data set. To the best of our knowledge, no publicly available data set exists with real nominal/emergency situations on which decision-making algorithm has been tested and published. Simulation could be used as a mean to generate collision data which could be used by the Intelligent Transportation System (ITS) community to compare algorithms. In addition, in this matter no standardize performance indicator exists to be able to compare algorithms between them. An interesting resolution for this year is the willingness of IEEE to start a range of recently recognized AV working groups. But IEEE is not alone, the diagram shown in Figure 3.3 illustrates the projects and norms either done by IEEE or other organizations. Under the reference IEEE P2846, the IEEE standardization organization has started working on the definition of "A Formal Model for Safety Considerations in Automated Vehicle Decision Making". The idea for industries and governments is to be able to eventually align with a common definition of what it means for an autonomous vehicle to make decisions that balances between safety and practicality. According to the IEEE, this approach is justified by the fact that the decision-making capacity of an on-

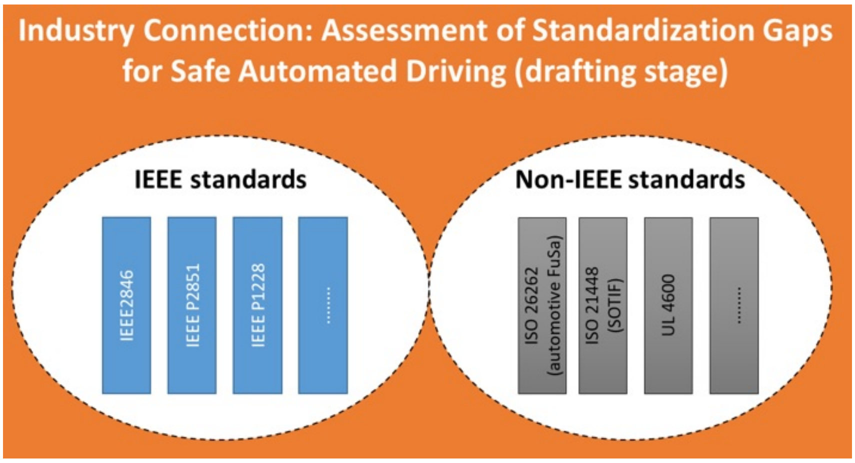


Figure 3.3: The approved working groups for AV Safety Standards by the IEEE [68]

board computer, with its set of artificial intelligence algorithms, is generally hidden from observation and constitutes a sort of "black box". This makes an objective comparison of the safety offered by different autonomous vehicles almost impossible. As some experts have already pointed out, the IEEE emphasizes that statistical evidence - such as the number of kilometers traveled, the frequency of human intervention or the hours of simulation - cannot capture all situations, especially those that the AV has never seen before. In concrete terms, the future IEEE P2846 standard aims to define a formal mathematical model based on rules related to vehicle decision-making using mathematical algorithms and discrete logic. The model will apply to the planning and decision making functions of an autonomous vehicle from levels 3 to 5 (cf. Figure 1.1). The model will be formally verifiable, via mathematical proof, will be technology-neutral and will be parameterizable to ensure the necessary customization at the level of individual jurisdictions. The standard will apply to specified scenarios and driving cases that do not eliminate all hazards, but that strike a balance between safety on the one hand and reasonable feasibility of application on the other.

A highly publicized contribution by Mobileye proposed a standardization of safety assurance and a formal model of safety by answering two main challenges: lack of safety guarantees, and lack of scalability. The lack of safety guarantees relies on answering the question what are the minimal requirements that every self-driving car must satisfy, and how can these requirements be verified. The second area of risk scalability concerns engineering solutions that result in huge costs will not be scalable to millions of cars, which will push interest in this field into a niche academic corner, and drive the entire field into what the authors called a "winter of autonomous driving". The combined issues leads the authors to propose a number of properties that answers to these challenges gathered in a framework called the Responsibility-Sensitive Safety (RSS) and that covers all the important ingredients of an autonomous vehicle: sense, plan and act. The RSS represent a rigorous mathematical model formalizing an interpretation of the law which is applicable to self-driving cars thus guarantees that from a planning perspective there will be no accidents which are caused by the autonomous vehicle. In addition, the framework has been designed such that it is not overly-defensive concerning the driving policy and efficiently verifiable in the sense that it can be proved that the self-driving car implements correctly the interpretation of the law. Jack Weast, Senior Principal Engineer at Intel has been appointed by the IEEE to lead the working group responsible for building the standard.

To get started, Intel will contribute with its Responsibility-Sensitive Safety (RSS) environment [191]. This framework, according to Intel, defines what it means for a machine to drive safely with a set of logically provable rules and appropriate responses prescribed for hazardous situations. In particular, the tool formalizes human notions of safe driving through transparent and verifiable mathematical formulas.

Many other works, such [61, 113, 161], try to make use of the RSS framework as a basis of their work. In [61], the authors propose to extend the RSS framework to include swerve maneuvers in addition to the standard brake maneuver available in the framework. These swerve maneuvers uses a kinematic bicycle model rather than the double integrator model of RSS. When vehicles are able to swerve and brake, it is shown that their required safe following distance at higher speeds is less than that required through braking alone. In addition, when all vehicles follow this new distance, they are provably safe.

3.8/ Conclusion

An important challenge in the field of risk assessment is to find the perfect balance between ensuring safety with all the imposed constraints: uncertainties, complexity and not being too conservative in the navigation. For these reasons defining risk assessment strategy based on four criteria is needed:

- Using multiple complementary criterion: this will allow to have redundancy in the assessment and improve the accuracy of the acquired information.
- Make use of the interaction aware formalism that takes into account the prediction and their interactions to assess the risk while keeping the complexity manageable.
- Handling noisy measurements.
- Safety verification and insure the generalization.

The next chapter will deal with the state of the art around decision-making strategies.

4

DECISION-MAKING IN AUTONOMOUS VEHICLES

Contents

Once the risk assessment made, the automotive system can start making the right decision. This chapter is dedicated to the related work on decision making for AVs. A classification is proposed of the different existing approaches.

4.1/ INTRODUCTION

The need of an efficient Decision-Making System (DMS) is rising as the ultimate challenge in nowadays research. The main reason is that decision-making is located at the highest level of the automotive architecture. An efficient DMS requires well-functioning sensors and perception, self-localization, precise maps, accurate interpretation and assessment of traffic situations to make safe and efficient decision. In order to execute these decisions, appropriate motion-planning and law level control functionalities (actuators control) should be mastered.

In addition, a DMS for autonomous driving according to Ulbrich et al., [207] requires:

- *Rapidity* in at least the planned decision for the driving maneuvers,
- *Coherency* as the decision making module should be consistent avoiding unnecessary switch in the plan,
- *Providentness* meaning the module should foresee how the situation will evolve after some time or some maneuvers and include it in the decision making,

- *Predictability* as the decision making should act while conforming to a human driver perception of safety and judgment for any situation.

Already in the early 90's, different systems for decision-making of autonomous vehicles were proposed under the designation behavior guidance or behavior decision. The ALVINN system proposed by Pomerleau [177], already discussed in section 2.4, learns through an artificial neural network to output steering angle to keep the vehicle on the road by taking images from a camera and a laser range finder as inputs. Few years later, Dickmanns et al. [63], proposed a model in which a rule-based structure would enable alternative maneuvers for the ego-vehicle triggered by special events recognized through vision. This is of course simpler then nowadays decision making systems but shows that the reflection was already ongoing.

Through the different DARPA Grand Challenge (2004, 2005 and 2007) (cf. section 2.3), multiple pioneering solutions have been proposed in multiple areas. But the decision-making as it is defined in nowadays research have been investigated only in the last edition: the DARPA Urban Challenge. The objective was to prove that navigation could be done in traffic when there were both moving intelligent vehicles and moving vehicles driven by humans. Competitors had to drive 97 km through urban environments, interact with other traffic participants and obey to the California traffic rules. This made it necessary to the competing team to use decision-making mechanism to guide the behavior of the controlled vehicles. Even if the DARPA Urban Challenge was a success, the conditions were still far from those of real-life scenarios. Most of the used decision-making strategies where deterministic, manually defined and significantly facilitated. The organizers frequently stopped vehicles to clear up traffic jam. Uncertainties in the other hand, could be mostly neglected or simplified. No objects blocked the visibility and instructed "human drivers" behave relatively predictably. However, the learned lessons from the Urban Challenges where very valuable and a lot of raised challenges are still subject of nowadays research. Rich from the expertise acquired from the Urban Challenges, simultaneously research both in academia and industry pursued their way at an accelerated rhythm with one goal, to reach a fully autonomously driving car.

Today's research in decision-making focuses mostly on finding global and robust solutions. Solutions that generalize to all situations while taking into account uncertainty, unpredictable situations and guarantee safety.

In what follows, we review the state of the art on decision-making for autonomous vehicles and propose a classification in three categories: Traditional approaches (cf. section 4.2), probabilistic approaches (cf. section 4.3) and learning based approaches (cf. section 4.4).

4.2/ Traditional approaches

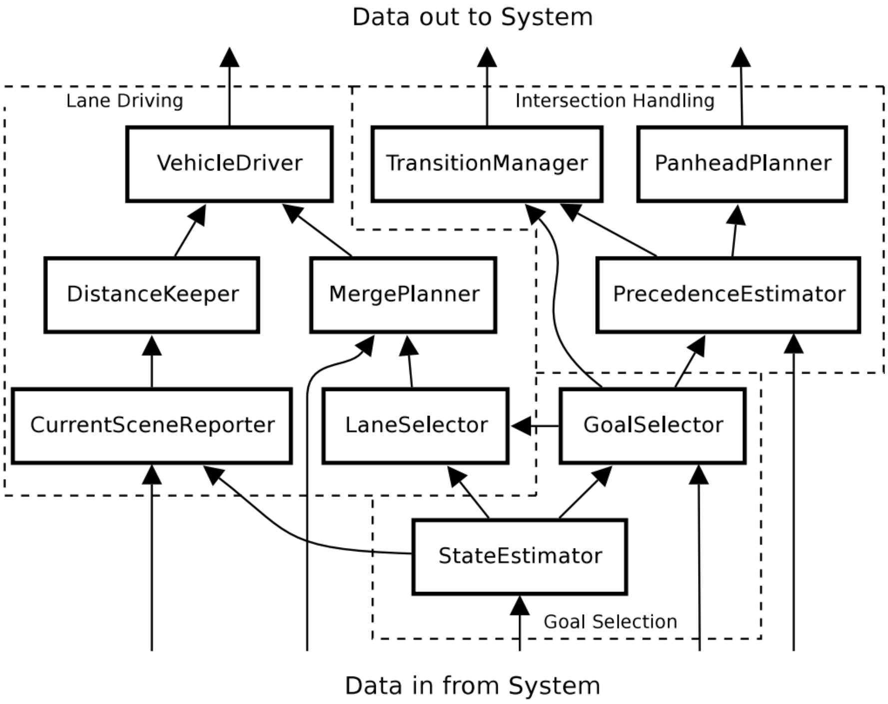
Traditional decision-making methods often involves building a system of rules and deducing the most suitable maneuver [186]. This approach lack the ability to generalize to unknown situations and to deal with uncertainties. In addition, when considering traffic scenarios, unexpected behaviors or perception modules failure that have not been considered during the construction of the system (which necessitates the addition of new rules and consequently) increases the complexity of the decision-making process.
The behavioral architecture (or the decision-making strategy) of the winning team of the DARPA Grand Challenge 2007 "The Tartan Racing team with Boss" (cf. Figure 4.1), is

based on the principle of identifying a set of driving contexts, each of which demands that the vehicle to focus on a reduced set of environmental characteristics. At the highest level of this design, the three used behaviors are: Lane Driving, Intersection Handling, and Achieving a Zone Pose. Achieving a zone pose behavior is meant for unstructured or unconstrained environments, including parking lots and jammed intersections. It is important to mention that the most common approach in the DARPA competitions was to use Finite State Machine (FSM) as decision-making strategy [131]. As an example Junior and Odin (the second and third ranked in the competition) used FSM to govern their vehicles' behavior. The FSM is given by a finite set of states in which the agent can be, and by the transitions between the states in response to some inputs. Based on the state of the vehicle in the world, the FSM searches for a suitable behavior (changing lane, making a U-turn) that makes the robot reach an objective checkpoint. Then, it either sends the controller the chosen trajectory that in return send steering and velocity command or send a message to the planner that the checkpoint cannot be reached. Most of these transitions are coded and tested by hand and thus prone to errors.
An even recent research using the concepts of state machine was the team working on the Bertha Benz Memorial Route [236]. However they structured the program flow of upcoming behaviors/decisions while using manually implemented architecture (cf. Chapter 2 and Figure 2.6). The authors proposed a hierarchical concurrent state machine as behavior generation method. Depending on the current driving situation, behavior generation formulates constraints (which is considered as the program flow) that arise from the current driving corridor, static obstacles, dynamic objects, and yield and merge rules. What we call decision-making is skipped in their approach and theses constraints are directly given to the trajectory planning modules (cf. Figure 4.2). In this case planning is used to make decisions in the specific situation.

Noh et al., [166] proposed a Bayesian Network for situation assessment that utilizes multiple complementary threat measures then the strategy decision component automatically determines an appropriate maneuver in a highway situation for collision-free, goal- directed behaviors via a hierarchical state machine.

In a similar fashion, multiple works focus on the definition of constraints that condition the trajectory planning module rather then defining a proper decision-making module. The trajectory planner chooses a trajectory considered as safe with respect to certain constraints related to the vehicles dynamic, the road geometry, the dimension of the vehicle or the occupancy of objects in the environment.
Model Predictive Control (MPC) for example sets the current control by anticipating future events using a model of the system dynamics. It is originally used as a control method but have been extended in the literature for decision making. In [165], it is presented a decision and control algorithm for lane change and overtaking maneuvers. The problem of deriving decisions regarding appropriate driving maneuvers i.e., selection of desired lane and velocity profile, on two-lane, one-way roads, is considered as a Mixed Logical Dynamical (MLD) system to be solved through MPC using mixed integer program formulation. The predictive controller allows full control of acceleration/deceleration as well as providing a decision variable regarding preferred lane at each time instant.

The advantage of the traditional approaches is their ability to be easily understandable and traceable for small problems. However, uncertainties and partial observability cannot be taken into account correctly in this kind of approach.



(a) Boss's High-Level Behaviors Architecture

Goal selection components	Drive down road	Handle intersection
State estimator: combines the vehicle's position with the world model to produce a discrete and semantically rich representation of the vehicle's logical position with the RNDF. ***Goal selector***: uses the current logical location as reported by state estimator to generate the next series of local goals for execution by the motion planner; these will be either lane goals or zone goals.	***Lane selector***: uses the surrounding traffic conditions to determine the optimal lane to be in at any instant and executes a merge into that lane if it is feasible. ***Merge planner***: determines the feasibility of a merge into a lane proposed by lane selector. ***Current scene reporter***: the current scene reporter distills the list of known vehicles and discrete obstacles into a few discrete data elements, most notably the distance to and velocity of the nearest vehicle in front of Boss in the current lane. ***Distance keeper***: uses the surrounding traffic conditions to determine the necessary in-lane vehicle safety gaps and govern the vehicle's speed accordingly. ***Vehicle driver***: combines the outputs of distance keeper and lane selector with its own internal rules to generate a so-called "motion parameters" message, which governs details such as the vehicle's speed, acceleration, and desired tracking lane.	***Precedence estimator***: uses the list of known other vehicles and their state information to determine precedence at an intersection. ***Pan-head planner***: aims the pan-head sensors to gain the most relevant information for intersection precedence decisions. ***Transition manager***: manages the discrete-goal interface between the behavioral executive and the motion planner, using the goals from goal selector and the gating function from precedence estimator to determine when to transmit the next sequence of goals.

(b) Components of the behavioral subsystem.

Figure 4.1: The decision-making in Boss system architecture (Image credit: [208])

4.3/ PROBABILISTIC APPROACHES

In difference to the methods shown in the previous section, where decision are either encoded by hand or directly nested in the planning, in this section we will discuss methods

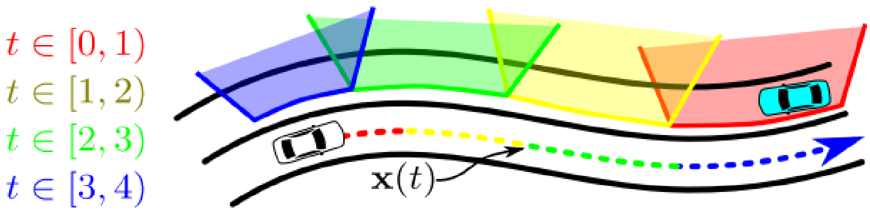


Figure 4.2: Constraints for an oncoming Object (cyan). The trajectory is constrained by polygons of corresponding color (Image credit: [236]).

based on a probabilistic definition. A special focus will be made on Bayesian decision making as it is among the main approach used in this Ph.D. work. Among the main used probabilistic framework for decision-making in the literature let us mention:

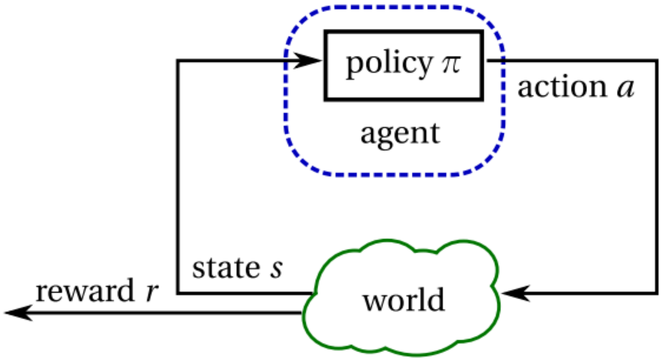
- Markov Decision Processes (MDP): Partially Observable MDP (POMDP), Mixed Observability MDP (MOMDP), Hidden Markov Model (HMM), semi-MDP (sMDP) [40, 82, 109, 144, 207].
- Bayesian Networks (BN): Two types are used the Dynamic Bayesian Network and the Decision Network (also called Influence Diagrams) [72, 186, 188] [1].

4.3.1/ Methods based on Markov Decision Processes (MDP)

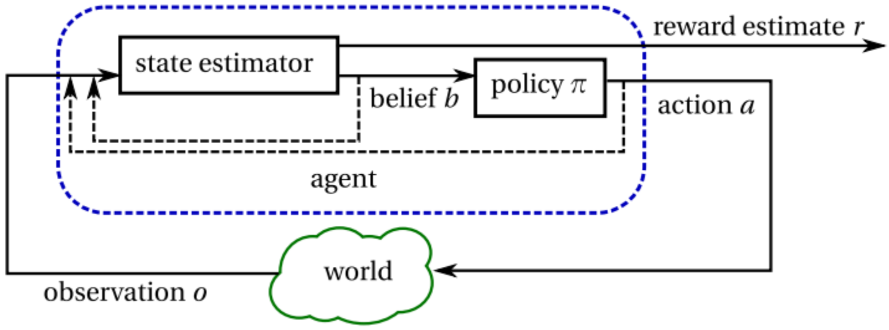
MDP is a discrete-time stochastic state transition system [28, 105]. An MDP is described as a tuple (S, A, T, R) where: S is the set of states; A the set of actions; T represents the transition probabilities of the system in other words it is the probability of ending in state x_t if the agent performs action a_{t-1} in state x_{t-1}; R is the reward that the agent will receive depending of the state of the system and the action taken, it states how *"desirable"* a state-action pair is for the agent.
It is used to model the sequential decision process of an agent acting in a dynamic environment with uncertain dynamics. In an MDP, the agent interacts with its environment - while taking the available information of the state of the world- by taking actions at discrete time steps. Upon taking such actions, the state of the world changes and make a transition and the agent receives a reward signal. Figure 4.3(a) shows the interaction of an MDP agent with its environment. It states that executing an action $a \in A$, given the system defined by a state$x \in X$, is what will be called a policy $\pi : x \rightarrow a$. The goal of such a problem is to find an optimal policy (sequence of actions) π^* that maximizes the expected reward over the time horizon T. The commonly applied approach to find an optimal policy is value iteration [28].

Mouhagir et al., in [157] proposed a method based on a MDP like model for trajectory planning with clothoid tentacles. The idea is to generate realistic trajectories with tentacles method and select the best tentacle regarding the MDP process. However the simulations was held with only static obstacles with no uncertainty consideration.

In [40] a theoretical approach is proposed for combining continuous world prediction by a DBN and discrete world semi-MDP planning. A semi-MDP allows actions that take varying amount of times to complete and is very useful in this work where the goal is to plan sequences of lane change maneuvers. The transition probabilities of the semi



(a) Markov Decision Process.



(b) Partially Observable Markov Decision Process.

Figure 4.3: Interaction of (a) an MDP agent or (b) a POMDP with the world (Image credit: [39]).

MDP are modeled using the DBN proposed by Gindele et al., [82] that accounts for the interactions between vehicles. The policy is then recalculated each time a new traffic situation is encountered. Which still raises the question of real-time applicability.

POMDP ([67], cf. Figure 4.3(b)) on the other hand, is an extension of an MDP to account for partial observability of states in a system. POMDP helps to introduce the idea of a belief $bel(x_t)$ of being in a state x_t at time t. A POMDP is represented by the tuple (S, A, T, R, Z, O) where (S, A, T, R) defined a fully observable MDP; Z a finite set of all measurements or possible observations at time t ; O is the observation function where for each action a and resulting state s, the observation function defines a probability distribution over the possible observations.

The objective of solving a POMDP is to find an optimal policy π^* which maximizes the expectation for the reward sum over the future time steps. However, despite the close relationship between POMDP and MDP, solving a POMDP is considerably more difficult than solving the corresponding MDP as the POMDPs computational complexity grows exponentially with the planning horizon. To overcome this issue, approximate solution methods have been proposed. Some of the solutions focus on solving POMDP models offline which means that the focus is not to calculate the best possible action for the current belief state but rather for every imaginable belief state. This restricts their applicability to small domains. In contrast, online approaches [109, 207] allows a calculation of a good policy at the current belied state of the agent. In what follows, some works using these methodologies for decision-making are presented.

In [207], online POMDP is used for decision-making for performing lane changes while driving fully automated in urban environments. An online POMDP to accommodate inevitable sensor noise to be faced in urban traffic scenarios. An ingenious way is proposed to keep the complexity of the POMDP low enough for real-time decision making while driving through a two steps algorithm as shown in Figure 4.4. A signal-processing

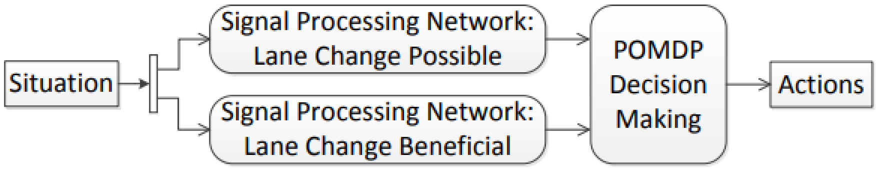


Figure 4.4: The two step algorithm for lane change decision-making. A network for signal processing is a graph in which each node represents a computation block and each edge represents a signal flow from one block to another. (Image credit: [207]).

network assesses the situation, whether a lane change is feasible or not, and whether a lane change is advantageous or not. The outputs of the signal processing networks are submitted to the POMDP decision-making algorithm.

Brechtel et al., in [41] in the other side presented a generic approach for decision-making under uncertainty through a continuous POMDP that can be optimized for different scenarios. The cornerstone of this work consist in considering uncertainties and finding the suited space representation for driving as the prevailing challenges for automatic decision-making. For this reason, the proposed POMDP automatically learns a suited representation depending on the specific given problem which is directly based in this paper on the pose and velocities of the involved road users.

We can see through the above mentioned works that the methodology in using POMPD differs in the way of solving them in terms of the used methodology, the complexity of the design and the problem space.

An Hidden Markov Model (HMM) is a temporal probabilistic model in which the state of the process is described by a single discrete random variable. The possible values of the variable are the probable states of the world. The work given in [135] proposed to use HMMs for the behavioral model with the aim of estimating the probability for a vehicle to perform one of its feasible behaviors. The behavioral model comprises two hierarchical layers, and each layer consists of one or more HMMs. The upper layer is a single HMM where its hidden states represent high-level behaviors, such as: move straight, turn left, turn right, or overtake. For each behavior in the HMM of the upper layer, there is a corresponding HMM in the lower layer which represents the sequence of state transitions of the behavior, for example *Overtake* has 4 hidden states: lane change, accelerate (while overtaking a car), lane change to return to the original lane, resume a cruise speed.

4.3.2/ Methods based on Bayesian Networks (BN)

The second well known family of probabilistic decision making is Bayesian Network. As stated before Bayesian Network for decision-making is used through two known extensions. The first one is Decision Network (DN) or Influence Diagram and the second is DBNs.

Bayesian Networks in summary are Directed Acyclic Graphs (DAG) in which each node corresponds to a random variable X_i connected by directed links called arcs. An arc from node X_i to node X_j signify that X_i is a parent of X_j.
For every variable X_i in the graph, with parents X_j Bayes theorem is applied to quantify the effect of the parents on the node and deduce the conditional probability distribution:

$$P(X_i|X_j) = \frac{P(X_i, X_j)}{P(X_j)} \tag{4.1}$$

The aforementioned conditional probabilities are summarized in a conditional probability table (CPT). For more details, we invite the reader to look at Annex A. The topology of the network specifies the conditional independence relationships between variables which makes Bayesian Networks by definition computationally tractable for reasonably small networks. This is one of the main advantages of Bayesian Networks.

BNs are used for probabilistic reasoning which is a method of representation of knowledge where the concept of probability is applied to indicate the uncertainty in knowledge. It is largely used in many industries and according to [181] BNs dominate AI research on uncertain reasoning and expert systems. It allows for learning from experience, and it combines the best of classical AI and neural nets.

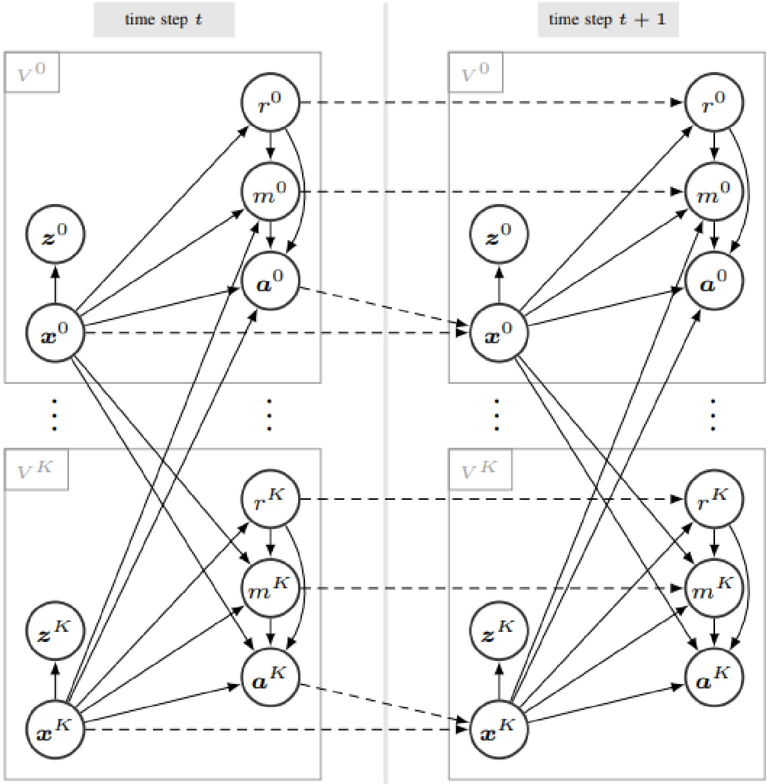
Decision Networks (DNs or Influence diagram) [106] combine BNs with additional node types for actions and utilities. DNs allow us to support probabilistic reasoning, decision-making under uncertainty for a given system and yield the capacity to incorporate multiple decision criteria. A detailed description of the main properties of Bayesian Networks are given in Appendix A.

In the field of decision-making for autonomous driving many use DNs. When modeling the DNs, most of the work chose to design the topology of the BNs with two main levels: the situation assessment level to infer the current situation state based on the risk assessment and the decision-making strategy to deduce the maneuvering decisions.

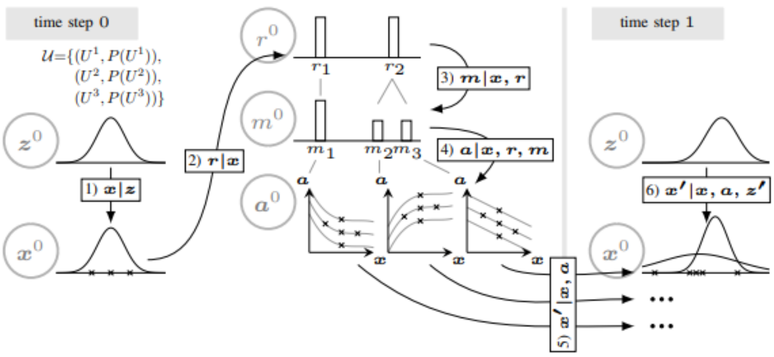
Using this formalism, Schubert in [186] uses DNs for lane change decision making. The paper presents a system that can perceive the vehicle's environment, assess the traffic situation, and gives recommendations about lane-change maneuvers to the driver. The situation assessment is done in the upper layer while using a well-known threat measure the *Deceleration To Safety Time (DST)* as a threat measure to assess the danger of the navigation lanes status. The lower layer is dedicated to decision making with one decision node for lateral maneuvers. The utility value is assigned manually for each combination of the traffic situation and the maneuver decision.

A DBN on the other hand is a Bayesian network that represents a temporal probability model. The system is modeled as series of snapshots or time slices each of which contains a set of random variables. DBNs generalize HMMS by allowing the state space to be represented in factored form, instead of a single discrete random variable.

DBNs are extensively used for maneuver intention, trajectory prediction and modeling the interaction between traffic participants as shown in Section 3.2.3 which makes them very suitable for decision-making. By making use of the interaction aware formalism, Schulz et al., [189] proposed a decision-making framework, which explicitly considers the intentions of drivers and the inter-dependencies between their future behaviors. The decision making process of the agent is divided into three hierarchical layers: which route it is going to take (route intention R_t), whether it is going to pass a conflict area at an intersection before or after another agent (maneuver intention M_t), and what continuous action A it is



(a) DBN showing the inter-dependencies between vehicles V_i.



(b) MM-UKF inference.

Figure 4.5: The decision-making (Image credit: [189])

going to execute. The proposed DBN is presented in Figure 4.5. It shows two consecutive time slices and the dependencies between random variables. The used inference algorithm is a Multiple Model Unscented Kalman Filter (MM-UKF) (cf. Figure 4.5). Each UKF represents the complete state space, i.e., kinematic state, route, maneuver, and action of all agents in the scene.

One of the pioneering contribution and a must cited example is the work of Forbes et al., [72] with the BATmobile. He proposed to use a Dynamic Probabilistic Network (DPN) shown in Figure 4.6. The used DPN resembles the definition of a DBN and contains nodes for sensor observations as well as nodes for predicting driver intentions, such as whether the driver intends to make a lane change or to slow down. The DPN is used as the basis for three separate decision-making approaches: dynamic decision networks which is the DPN extended with actions node and utility function for each time slice, hand-coded policy representations through a decision tree and supervised learning and reinforcement learning methods for solving the full POMDP. Even though Forbes et al., deduced that avoiding manual programming and considering partial observability improves the results of their proposed solution were ahead of their time given the present state of technology.

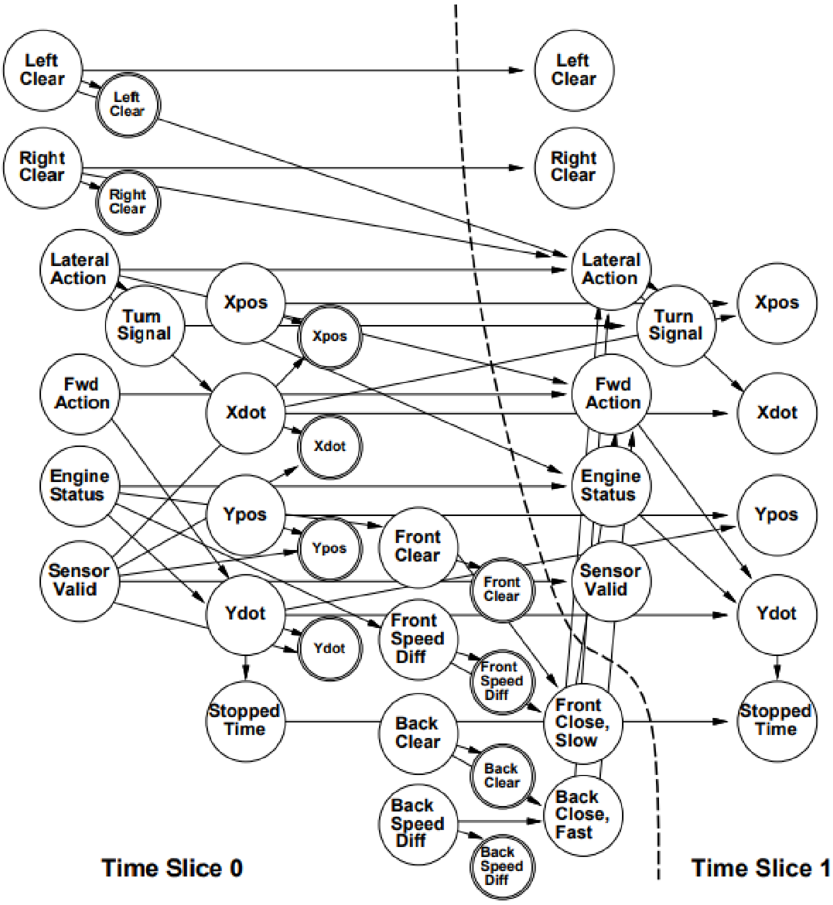


Figure 4.6: The BATmobile dynamic probabilistic network for one vehicle (Image credit: [72]).

4.4/ LEARNING-BASED APPROACHES

In this section we will focus only on the methods developed specially for decision-making using learning approaches as end-to-end techniques have been detailed in Chapter 2, section 2.4.

To overcome the drawbacks raised from other methods, human-like decision making based on Deep Learning have sprung up. Contrary to the aforementioned methods, these methods recognize human personality and social intelligence and does not fully focus on the "correctness" [192] as they learn from real driving scenarios.

According to Google, accidents that occurred with their autonomous vehicles can also be used as a valuable experience for the self-driving system [233]. On this basis, several neural network have been proposed for the decision making strategies. In [143] a decision-making system is presented, the main novelty lies in the human-like thinking ability integrated in the neural network. In addition to the mentioned methods, Reinforcement Learning (RL) have also been used. For instance, the Inverse Reinforcement Learning (IRL) model [132] is used to get the individual driving style of traffic participants and plane the safest trajectory. In a similar fashion, Q-learning algorithm have been applied for lane change scenarios [231]. Ngai et al., in [162] proposed a reinforcement learning algorithm of lane change decisions for highway driving. By making use of Q-learning for determining action decisions, seven different goals are considered among which: lane changing, collision avoidance, and lane following, as the author believes that by explicit-ing the goals, the problem can be better solved.
The mentioned deep learning methods do offer great advantages in terms of flexibility and scope of utilization. However, the main drawback is that no one can analytically ensure that the corresponding output for these systems will tend towards always an acceptable safe solution. Ultimately, the cost incurred is not worth the risk. For this reason some works choose to combine deep learning approaches with safety verification methods to guarantee safety of the decisions. Mirchevska et al.,in [154] presents a reinforcement learning-based approach, that is combined with formal safety verification to ensure that only safe actions are chosen at any time. The deep reinforcement learning agent learn to

Approach	Reference	Prediction	Uncertainty	Offline/Online	Space	Generalization
Traditional approaches	Manual Decision-Making e.g., FSM [131]	X	X	Offline	Discrete	- -
	Trajectory Planning e.g., Bertha Benz Memorial [236]	$\checkmark$	$\checkmark$	Online	Continuous	++
Probabilistic approaches	Decision Networks e.g., Schubert et al., [186]	($\checkmark$)	Uncertainty in the states	Offline	The states are either discrete or continuous, it uses a utility value for decision	+
	POMDP e.g., Brechtel et al., [41]	$\checkmark^{a,b}$	$\checkmark$	Offline	Continuous	+
	DBN e.g., Schulz et al., [189]	$\checkmark^{a,b}$	In the states of the DBN and on the measurements	Online	Continuous	++
Learning-based approaches	Reinforcement Learning [162]	($\checkmark$)	($\checkmark$)	Offline	Discrete	-

Table 4.1: Comparison of decision-making approaches for AVs. "($\checkmark$)" means that the feature is not supported in the original work but can easily be integrated. $\checkmark^a$ means the prediction considers interaction between traffic participants. $\checkmark^b$ means that long time horizon prediction is considered. Offline/Online means whether the system find the best possible maneuver to be executed in the current situation or during an offline training phase.

drive as close as possible to a desired velocity by executing reasonable lane changes on highways.

However, this doesn't resolve the main drawback of deep neural networks concerning the computational complexity needed in the learning phase and the fact that these methods do not have any analytical formulation, which leads to non-provable outputs.

4.5/ CONCLUSION

This chapter discussed the state of the art algorithms in the domain of decision-making of AVs. Table 4.1 summarizes the main characteristics of the aforementioned decision-making algorithms. It illustrates a comparison between performances of the traditional, the probabilistic and the learning-based approaches.

The most important aspects in a decision-making framework is its ability to solve any situation, consider uncertainty and unexpected situation while finding the right balance between accuracy and computational expenses. Probabilistic approaches come out to be the best regarding the defined characteristics as it have the potential to consider the nature of the stochastic dynamics of a traffic environment, be able to account for uncertainties through well-known probabilistic algorithms and consider for present and future interactions between participants.

II

Proposed multi-controller architecture for safe autonomous navigation under uncertainties

5

PROBABILISTIC MULTI-CONTROLLER ARCHITECTURE (P-MCA)

Contents

The state of the art review on automotive safety systems and control architectures conducted in chapter 1 led to the conclusion that hybrid architectures have the utility of dealing with the limitations raised from classical and IA based architectures, if the good equilibrium between both approaches is found. In this aim, this chapter presents a hybrid framework for handling navigation of autonomous vehicles through a Probabilistic Multi-Controller Architecture (P-MCA). Section 5.1 details the main assumption and background work. Section 5.2 introduces the principles and the main functionalities behind the proposed P-MCA and justify the choices made concerning the approach used with a focus on the motion planning modules for road-way navigation (cf. section 5.2.2). The rest of the chapter is dedicated to the risk management strategy and to the proposed criteria for anomaly detection. Throughout the chapter demonstrative examples are shown to validate each part.

5.1/ Main control and navigation strategy assumptions

In order that this dissertation becomes at maximum self-contained, the main assumption and background work will be given in this section. Indeed, the proposed Probabilistic Multi-Controller Architecture (P-MCA) is the immediate extension of an already developed Multi-Controller Architecture (MCA) for mono- and multi-vehicle navigation in cluttered environments proposed in [214, 216, 217] and [218]. In this section, the main components of the MCA will be given.

5.1.1/ Multi-Controller Architecture (MCA)

As detailed in section 2.3 of chapter 2, the main aim of multi-controller architectures is to have a bottom-up construction of the principal functionalities of autonomous vehicle navigation. In other words, it aims at decomposing an overall complex task into a multitude of sub-tasks to achieve [8]. In other words, it aims at decomposing an overall complex task into a multitude of sub-tasks to achieve [8]. The definition of Advanced Driver Assistance Systems (ADAS) is close in a certain manner to what we claim in this dissertation though the use of multi-controller architectures (MCA). Indeed, this implies the development of appropriate reliable elementary controllers (e.g., obstacle avoidance, target reaching/tracking) and specific mechanisms to manage the controllers' interactions.

The MCA shown in Figure 5.1 [214, 216, 217, 218] constitutes the starting block of our work. It is proposed to manage interactions among elementary behaviors while guaranteeing the stability of the overall control [6]. The architecture is governed by two elementary behaviors: Target reaching and obstacle avoidance that will be detailed in the following subsections 5.1.3 and 5.1.2 respectively. At each sample time one of them is activated by a Hierarchical action selection mechanism according to the perceptive fea-

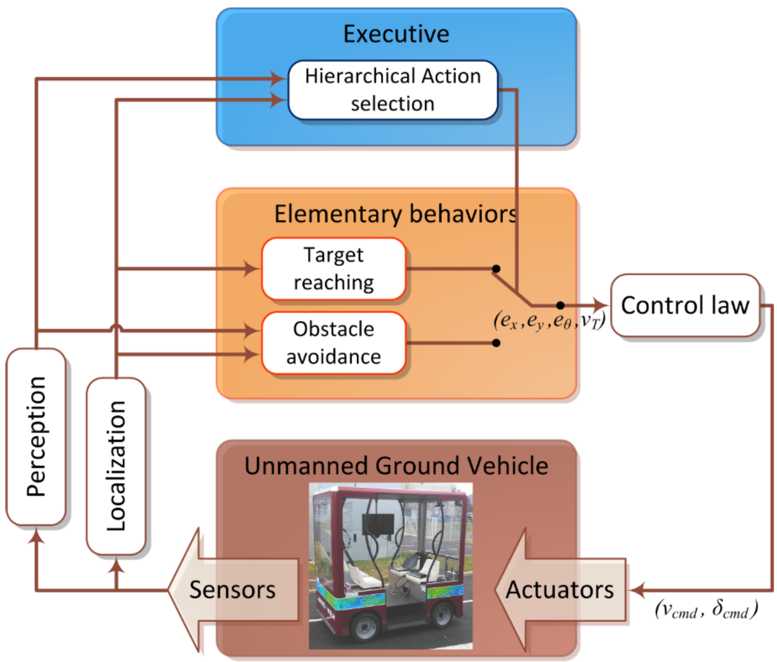


Figure 5.1: Vilca's Control Architecture [Image credit: [219, Chapter 3]]

tures. Each vehicle's controller is constituted by a dedicated homogeneous set-points defined by a pose (x_T, y_T, θ_T) and a velocity v_T. It will be highlighted that this set-point formulation is generic enough to define an important number of the vehicle's behaviors. It is important to emphasize that the navigation is performed while tracking these set-points and not in a trajectory following behavior. It is to be noted that, once the set-points are defined, at each sample time, it is important to have reliable control laws to reach/track these assigned set-points. To do that, a unique stable control law defined in section 5.1.5 shared by the two set-point blocks will be used to stabilize the errors to zero and thus allow us to reach/track these assigned set-points.

5.1.2/ Obstacle Avoidance based on Elliptic Limit-Cycle (ELC)

Before making the focus on the used method for set-points extraction, let us introduce in short the used obstacle avoidance method.
The navigation task of a vehicle must be achieved while avoiding static and dynamic obstacles. These obstacles can have different shapes and the avoidance can be done using different obstacle avoidance techniques like potential field methods[126], Vector Field Histogram [36] or limit-cycles trajectories.
This last method correspond to specific trajectories that are defined according to a circular [7, 128] or an elliptic periodic orbit [11] (called the ellipse of influence (EI)), where converge all the trajectories starting inside or outside the given orbit. The advantage in using elliptic orbits is that an ellipse fits better many obstacles shape e.g., walls or cars, than circles. These periodic orbits if well-dimensioned (far enough from any obstacle) and the resulting trajectory accurately followed, the avoidance of any given obstacle is guaranteed (cf. Figure 5.2).

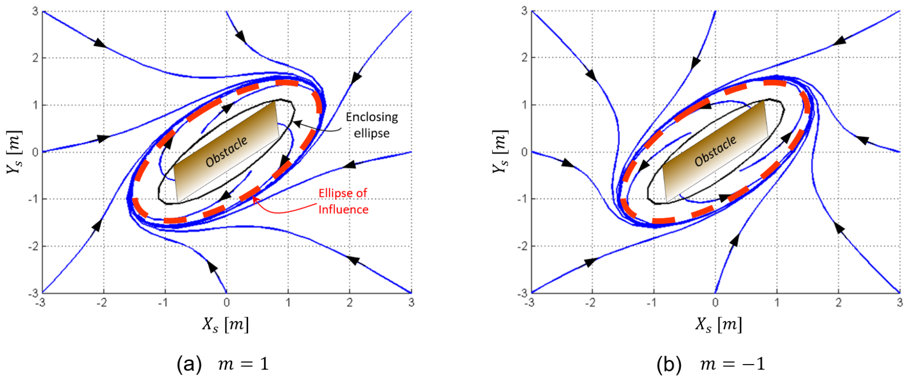


Figure 5.2: Clockwise ($m = 1$) and counter-clockwise ($m = -1$) shape for Elliptic Limit-Cycles (ELC)

These ELC trajectories [10] are defined according to a set of differential equations :

$$\dot{x}_s = r(By_s + 0.5Cx_s) + \mu x_s(1 - Ax_s^2 - By_s^2 - Cx_s y_s) \tag{5.1}$$
$$\dot{y}_s = -r(Ax_s + 0.5Cy_s) + \mu y_s(1 - Ax_s^2 - By_s^2 - Cx_s y_s) \tag{5.2}$$

with (x_s, y_s) corresponds to the position of the vehicle according to the center of the ellipse of influence; $r = \pm 1$ according to the avoidance direction (clockwise (+) or counter-clockwise (-) respectively, cf. Figure 5.2); $\mu \in \mathbb{R}^+$ a positive constant value which allows us to modulate the convergence of the ELC. The convergence is as slow as μ is smaller, which allows us also to obtain smoother ELC. The variables A, B and C are given by:

$$A = (\sin(\Omega)/b_{lc})^2 + (\cos(\Omega)/a_{lc})^2 \tag{5.3}$$
$$B = (\cos(\Omega)/b_{lc})^2 + (\sin(\Omega)/a_{lc})^2 \tag{5.4}$$
$$C = (1/a_{lc}^2 - 1/b_{lc}^2)\sin(2\Omega) \tag{5.5}$$

where a_{lc} and b_{lc} characterize respectively the major and minor elliptic semi-axes and Ω gives the the ellipse of influence orientation. Figure 5.2 shows the shape of equations (5.1) and (5.2) when $r = 1$ and -1 respectively. They show the direction of trajectories (clockwise or counter-clockwise) according to (xs, ys) axes. The trajectories from all points (xs, ys) of X, Y reference frame, including inside the the ellipse of influence, move towards the the ellipse of influence.

Obstacle avoidance algorithm based on the limit-cycle approach have been used in [11, 215] for mobile robot navigation in cluttered environment. This algorithm guarantees the avoidance of the obstacle by the robot while performing smooth trajectories.

Concerning the definition of the set-point configurations, they are taken within the generated ELC trajectories.

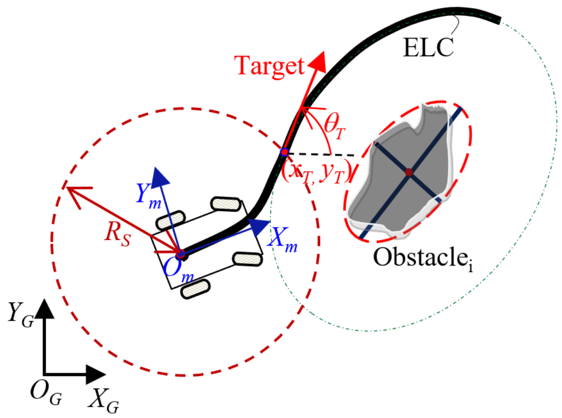


Figure 5.3: Set-points definition based on local planned path [Image credit: [10]]

In this first case, when the environment is not very well known, it is better to navigate reactively. In that situation, the current ELC takes as initial configuration, and at each sample time, the current robot configuration. The target set-point is given by (cf. Figure 5.3):

- A position (x_T, y_T) corresponding to the intersection between the circle (which has as origin the origin of the reference frame X_mY_m and as radius R_S) and the planned ELC.
- An orientation θ_T corresponding to the tangent to the ELC w.r.t. X_GY_G reference frame at the intersection point (x_T, y_T). If $R_S = 0$, the robot has to apply only an orientation control. Indeed, since the robot is already on the current computed ELC, the robot has only to control its heading w.r.t. θ_T.
- A velocity v_T which could be constant or variable indifferently.

5.1.3/ Target Reaching

The second identified case corresponds to the one where a global path is already defined using for example an ELC [9]. In this case, it is enough for the robot to follow the path as accurately as possible without modifying its initial planning. In that situation, a Frenet reference frame is used [183] to extract the robot's set-points. The target set-point, at each sample time is given by (cf. Figure 5.4):

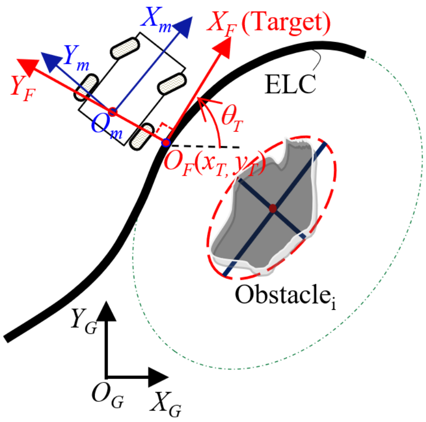


Figure 5.4: Set-points definition based on global planned path [Image credit: [10]].

- A position (x_T, y_T) corresponding to the closest position in the pre-planned path w.r.t. the origin of the reference frame X_mY_m. (x_T, y_T) point corresponds to the origin of Frenet reference frame X_FY_F.
- An orientation θ_T corresponding to the tangent of the path w.r.t. X_GY_G reference frame.
- A velocity v_T which could be constant or variable indifferently.

5.1.4/ Control variables

The previously detailed blocks are responsible for generating the control variable to drive the robot toward the defined target configuration. These control variable are related to the relative pose between the robot and the target and they are represented by the error state (e_x, e_y, e_{theta}) shown in Figure 5.5:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_T - x \\ y_T - y \\ \theta_T - \theta \end{bmatrix} \tag{5.6}$$

The error function e_{RT}, representing the error related to the robot's position (x, y) w.r.t the target's orientation θ_T, is added to the canonical error state eq. (5.6) (cf. Figure 5.5). Let us first define d and θ_{RT}. d and θ_{RT} are respectively the distance and the angle between the position of the target and the robot and they are defined as:

$$d = \sqrt{(x_T - x)^2 + (y_T - y)^2}$$
$$\theta_{RT} = \arctan\left(\frac{y_T - y}{x_T - x}\right) \quad \text{if } d > \xi$$
$$\theta_{RT} = \theta_T \quad \text{if } d \leq \xi$$

where ξ is a small positive value ($\xi \approx 0$) The error e_{RT} is defined as (cf. Figure 5.5):

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

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

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

where $\tan(\theta_{RT} - \theta) = e_y e_x^{-1}$ (cf. Figure 5.5). Hence, e_{RT} allows to consider an additional orientation error w.r.t. e_x, e_y and e_θ, e.g., when $e_\theta = 0$ then $e_{RT} = -e_y/e_x$. The stabilization of this error allows to decrease the lateral distance d_l to zero eq. (5.9) (cf. Figure 5.5), and to always have the robot in the wake of the target.

These error state and the velocity $\left(e_x, e_y, e_\theta, v_T\right)$ are given as input to the control law (cf. subsection 5.1.5 that allows the convergence of this variable in the purpose of guiding the robot towards the target.

5.1.5/ Uniform Control law

Before presenting the used control law, it is important to know the vehicle's model.

The used vehicle corresponds to a tricycle vehicle [62] modeled according to the well-known kinematics model given by equation 5.7.

$$\begin{cases} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= v\tan(\gamma)/l_b \end{cases} \tag{5.7}$$

where (x, y, θ) is the posture (configuration state) of the vehicle at the point O_m (origin of the local reference frame $X_m Y_m$ linked to the vehicle Figure 5.5), γ is the orientation of the equivalent front wheel Figure 5.5, v is the linear velocity of the vehicle at O_m and l_b is the vehicle's wheelbase. v and γ are the two control inputs of the vehicle (cf. equations 5.11 and 5.12 respectively).

According to Figure 5.5, w_b corresponds to the track width of the vehicle and I_{cc} the instantaneous center of curvature of the vehicle trajectory. The radius of curvature r_c is given by:

$$r_c = l_b / \tan(\gamma) \tag{5.8}$$

and $c_c = 1/r_c$ is the curvature of the vehicle trajectory.

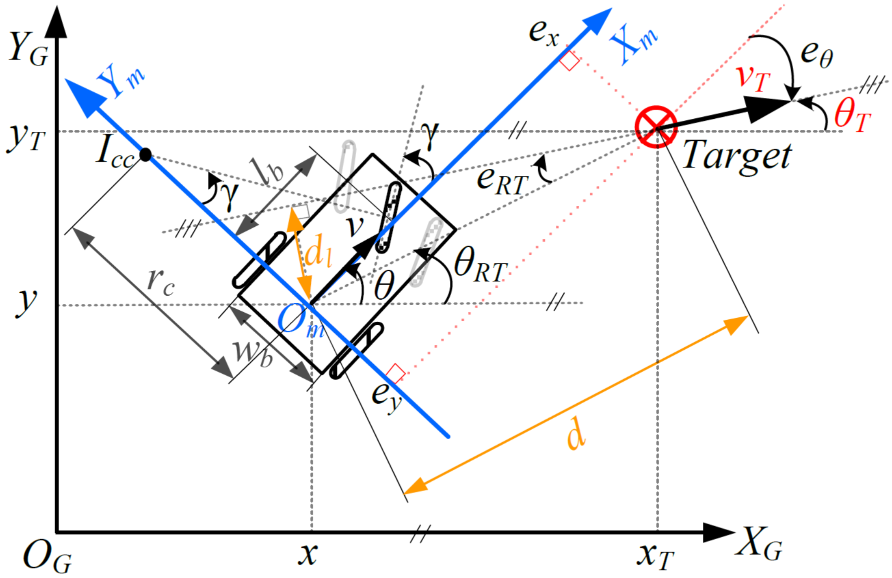


Figure 5.5: Vehicle's and target's configuration in global (X_G, Y_G) and local (X_m, Y_m) reference frames, and the control variables. [Image credit: [214]]

The used control law [214] aims to drive the vehicle toward specific targets (static or dynamic) in the environment. At each sample time the tracked target is defined by a posture (x_T, y_T, θ_T) and a velocity v_T (this velocity could be $= 0$ if the target is static). As it was shown in subsections 5.1.3 and 5.1.2, different vehicles behaviors are described with a uniform way where the vehicle has to reach/follow/track a specific target set-points. In order that this paper becomes at maximum self-contained, let us give in summary the main elements of synthesis, using a Lyapunov formulation, the used control law [214]. The adopted Lyapunov function V is given by equation (5.9) Figure 5.5:

$$\begin{aligned} V =& \frac{1}{2}K_d d^2 + \frac{1}{2}K_l d_l^2 + K_o[1 - \cos(e_\theta)] \\ =& \frac{1}{2}K_d d^2 + \frac{1}{2}K_l d^2 \sin^2(e_{RT}) + K_o[1 - \cos(e_\theta)] \end{aligned} \tag{5.9}$$

where the initial values of e_{RT} and e_θ must satisfy the following initial conditions:

$$e_{RT} \in]-\pi/2, \pi/2[\quad \text{and} \quad e_\theta \in]-\pi/2, \pi/2[\tag{5.10}$$

The Lyapunov function (5.9) is therefore a function of three parameters which depend on: the distance d between the target and vehicle's position; the distance d_l from the vehicle to the target line (line that passes through the target position with an orientation equal to the target orientation), this term is related to the Line of Sight and Flight of the target [196]; and the orientation error e_θ between the vehicle and the target.

The desired linear velocity v and the front wheel orientation γ of the vehicle which allows to asymptotically stabilize the error vector $(e_x, e_y, e_\theta, (v - v_T))$ toward zero (permitting therefore to have $\dot{V} < 0$) are given by:

$$v = v_T \cos(e_\theta) + v_b \tag{5.11}$$

$$\gamma = \arctan(l_b c_c) \tag{5.12}$$

where v_b and c_c are defined by:

$$v_b = K_x \left[K_d e_x + K_l d \sin(e_{RT}) \sin(e_\theta) + K_o \sin(e_\theta) c_c\right] \tag{5.13}$$

with:

$$\begin{aligned} c_c =& \frac{1}{r_{c_T} \cos(e_\theta)} + \frac{d^2 K_l \sin(e_{RT}) \cos(e_{RT})}{r_{c_T} K_o \sin(e_\theta) \cos(e_\theta)} + K_\theta \tan(e_\theta) \\ &+ \frac{K_d e_y - K_l d \sin(e_{RT}) \cos(e_\theta)}{K_o \cos(e_\theta)} + \frac{K_{RT} \sin^2(e_{RT})}{\sin(e_\theta) \cos(e_\theta)} \end{aligned} \tag{5.14}$$

K $= (K_d, K_l, K_o, K_x, K_\theta, K_{RT})$ is a vector of positive constants defined by the designer. Accurate analysis of this stable and efficient control law is given in [215] and [214].

In the following sections, the proposed Probabilistic Multi-Controller Architecture (P-MCA) an improved version of the MCA adapted to road-way navigation is detailed, .

5.2/ The Probabilistic multi-controller architectures (P-MCA) for road-way navigation

5.2.1/ The P-MCA main functionalities

The main functionalities developed in our work aims in priority to improve the reliability of the overall MCA (detailed in section 5.1) in order to deal efficiently with uncertainties characterizing road-way navigation, more precisely to perform overtaking maneuvers in dynamic environments and introduce a high-level decision-making strategy to supervise these interactions. The Probabilistic Multi-Controller Architecture (P-MCA) shown in Figure 5.6 (initially motivated in [1]) has been proposed around several complementary modules to plan/control and to assess and manage the risks of autonomous navigation in dynamic and uncertain environments. These blocks and their main functionalities are summarized below.

Perception and localization module The perception and localization block are in charge of providing the important features of the environment needed for navigation such as the pose of the perceived obstacles, the number of lanes and the lane marking, and the sensory uncertainty. We assume redundant and reliable hardware, allowing us to ignore hardware faults for the subsequent discussion. This module will not be further detailed as it is out of the scope of this work.

Route planning module The route planning module gives selected sequence of way-points through the road network. It also uses a closed loop from the decision-making to specify a short term goal within the road network that is defined always ahead of the ego-vehicle. This information is used when defining the obstacle avoidance strategy as will be explained in Section 5.2.2.3.

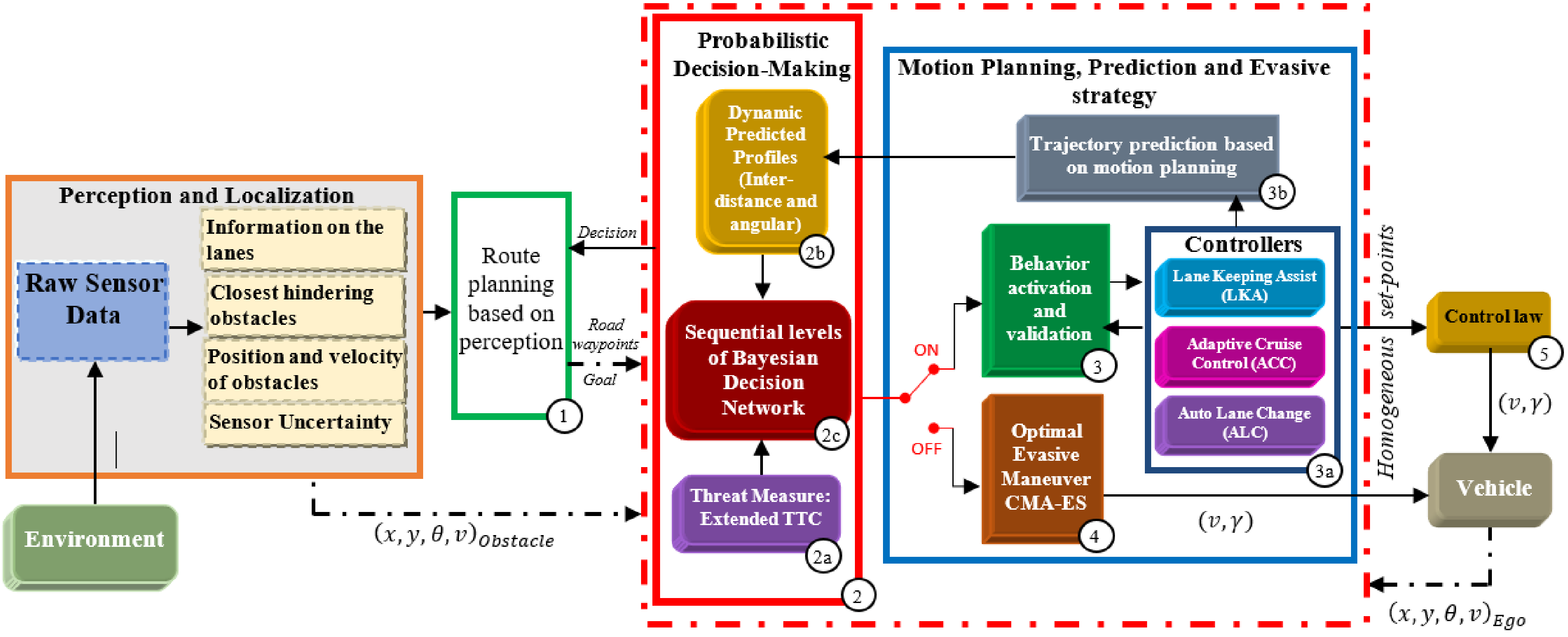


Figure 5.6: Probabilistic Multi-Controller Architecture (P-MCA) for autonomous vehicles' navigation. The highlighted box in red discontinue line corresponds to the focus and the main contributions of the Ph.D thesis.

Probabilistic Decision-Making module and its safety assessment and verification criteria Once the perceptive and route planning features are defined, an appropriate decision-making strategy for autonomous navigation has to be adopted that takes into account several aspects, such as: traffic rules, passenger safety and measurement uncertainty of perceptive modules.
The decision-making relies on a data-driven approach and consist of a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) (block 2c, detailed in chapter 6) that utilizes multiple complementary threat measures (block 2a and 2b, detailed in section 5.3)) to propose discrete actions and derive the appropriate maneuver in a given traffic situation. The SDN-MSV is also responsible for: having a safety retrospection and verification over the current maneuver risk and advising appropriate evasive action autonomously from dynamic obstacles. Unlike other AI approach like neural networks that needs extensive amount of data and learning time, the proposed decision-making strategy has a short response time given that Bayesian Networks (BNs) are computationally tractable due to the exploitation of conditional independence relationships. On the other hand, finding a realistic mathematical model that is able to understand the environment and its dynamic and make real-time decisions is not a simple task. BNs are also able to handle uncertainty that may arise from uncertain observations. In contrast, neural networks can solve problems with uncertainty however massive and even quasi exhaustive data configuration should be available (cf. Table 2.1).

Motion Planning, prediction and evasive strategy module The common task for autonomous vehicles after the decision-making is to apply the decided maneuver by determining a nominal trajectory to perform lane changes and other maneuvers while taking into consideration any constraints or traffic conditions that are known at the time of planning. This is performed in block 3a through the elementary controllers and are detailed in section 5.2.2). The controllers relies on the proposed architecture on the background work detailed in section 5.1 and adapted to road-way navigation. At each sample time, one of the controllers is activated through the behavior activation process (block 3) based on the high level decision-making.
The maneuver must therefore be aborted automatically in case of any unexpected approaching objects such as other objects and road users entering the planned course of the vehicle. The system must define an evasive strategy to determine an alternate route, i.e., the emergency trajectory or low-level control (block 4, cf. section 6.4) which the AV should adopt instantly to avert an accident. For this purpose, an appropriate evasive strategy is proposed and is detailed in Section 6.4 that combines both the Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) and a Covariance Matrix Adaptation Evolution Strategy (CMA-ES).
The trajectory prediction in the other side is directly used as input to the proposed safety verification strategy detailed in section 5.3.3.

Control law module The used control law is the one detailed in section 5.1.5 that aims to drive the vehicle toward specific targets in the environment. The control law is synthesized for a tricycle model based vehicle using Lyapunov synthesis and is detailed in section 5.1.5.

All of this modules are gathered in what we called a Probabilistic Multi-Controller Architecture (P-MCA) that effectively links model-based approaches and Artificial Intelligence

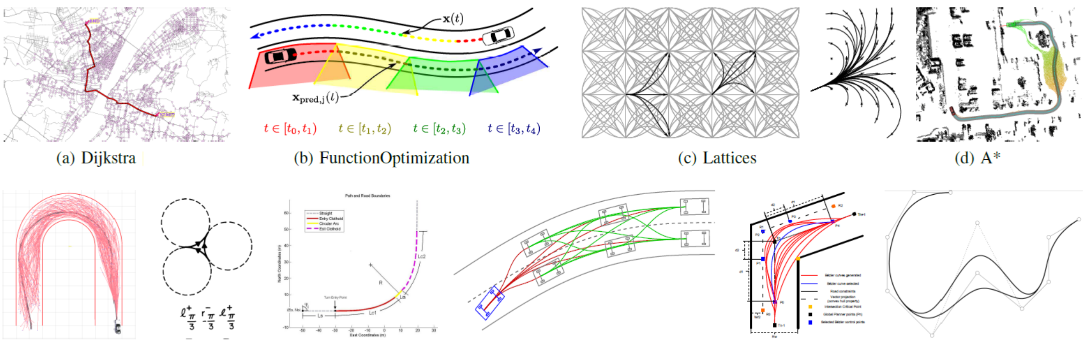
(AI) developments for intelligent vehicles navigation while guaranteeing the safety of maneuvers even in presence of uncertainty. The readers may refer to Chapter 2 for hybrid architectures review and precedent authors work [3].

5.2.2/ Elementary controllers

The controllers relies on the proposed architecture on the background work detailed in section 5.1 but the reasoning is fully adapted to road-way navigation. The control in the other side is used as defined in subsection 5.1.5.

During autonomous navigation in road-way, vehicles perform a multitude of tasks while guaranteeing the smoothness and the safety of the the AV's trajectories. Among these stand out the following main elementary behaviors performed by the AV:

- **Lane Keeping Assist (LKA)** where lateral set-points are defined to follow the center of the current lane and the longitudinal ones are to maintain a desired velocity.
- **Adaptive Cruise Control (ACC)** in a car following behavior (using the Lane Keeping Assist to stay in the center-line of the lane) for driving with desired velocity while maintaining a safety distance with vehicles ahead.
- **Automatic Lane Changing (ALC)** which is activated under either directional or obstacle constraints when for example overtaking a slower vehicle and returning back to the original lane.



(a) Dijkstra (b) FunctionOptimization (c) Lattices (d) A*
(e) RRT (f) Line&Circle (g) Clothoid (h) Polynomial (i) Bezier (j) Spline

Figure 5.7: Review of motion planning algorithms in the literature [Image credit: [85]]

Many works in the literature surveys motion planning algorithms for self-driving cars [55, 85, 167], and different techniques are proposed for each algorithms. A number of classification have also been proposed. For example [85] proposed four algorithm groups for motion planning techniques applied in automated driving scenarios: Graph search based planners e.g., Dijkstra or A*, Sampling based planners e.g., Rapidly-exploring Random Tree (RRT), Interpolating curve planners e.g., Clothoid [157] or Bézier curves, and Numerical optimization approaches e.g., Ziegler et al., [235]. A detailed comparison is proposed in [55, 85] and some examples are shown in Figure 5.7.

In our work, at each sample time, one of the above mentioned behaviors is activated based on a behavior activation process (block 3, cf. Figure 5.6) that is in its turn is based

on: the high-level decision-making (cf. chapter 6), the maneuver safety verification (cf. section 6.3.3) and defined deterministic criteria regarding the precedent task achievement. The latter, is used for example, to verify the completion of a lane change maneuver that is considered achieved if the vehicle reaches the center-line of the left lane and remain steady around this line (cf. subsection 5.2.2.3).
The corresponding selected controller generates homogeneous *"dynamic"* target set-points X_T defined by a pose (x_T, y_T, θ_T) and a velocity v_T (subsections 5.1.3 and 5.1.2) that is used in its turn to compute an error state (e_x, e_y, e_{theta}) (cf. subsection 5.1.4) input to the control law (cf. subsection 5.1.5).

The following sections will detail the adopted strategy regarding the elementary controller to adapt the background work defined in subsections 5.1.3 and 5.1.2 for the above defined behaviors for road-way navigation.

5.2.2.1/ Lane Keeping Assist (LKA) and Adaptive Cruise Control (ACC) based on target reaching

Inspired from the target reaching behavior defined in subsection 5.1.3, the LKA and ACC controllers correspond to a case where a global path is already defined to be the center-line of the lane to follow and in this case it is enough for the vehicle to follow this path as precisely as possible (cf. Figure 5.8). The dynamic target set-points are then extracted from the center-line of the lane where the vehicle is localized using a Frenet reference frame [9] (cf. Figure 5.8). The target set-point X_T is defined as:

- a position (x_T, y_T) corresponding to the closest position in the path with an offset curvilinear distance R_{curv}, w.r.t. the origin of the vehicle reference frame $X_m Y_m$. The

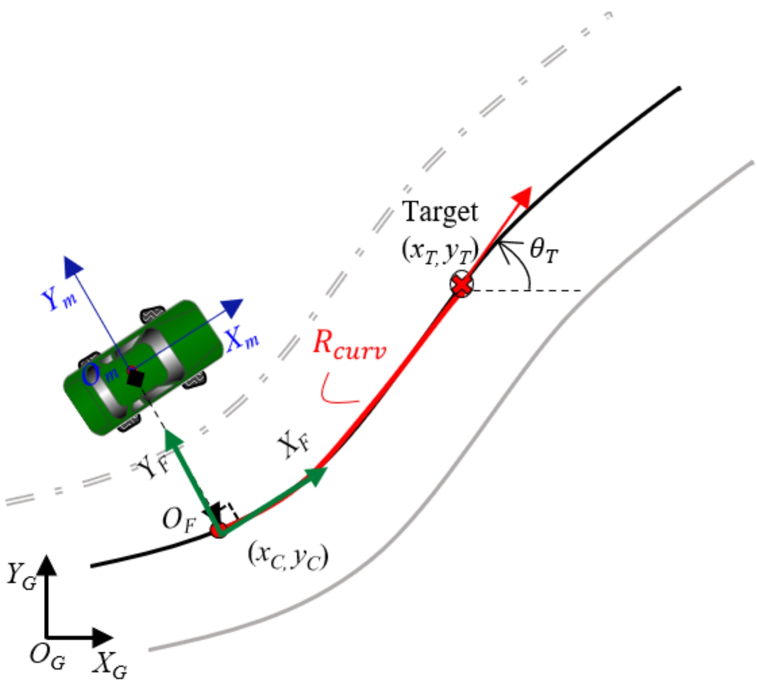


Figure 5.8: Homogeneous set-points definition based on dynamic target tracking: LKA and ACC based on Frenet reference frame

point corresponding to the closest position in the path (x_C, y_C) corresponds to the origin of the Frenet reference frame $X_F Y_F$.

- an orientation θ_T tangent to the defined path at (x_T, y_T).
- a velocity v_T corresponding to the velocity of the ego-vehicle so that the dynamic target set-point is always ahead at a defined offset R_{curv}.

R_{curv} is fixed empirically while taking into account the velocity of the ego-vehicle, the length of the wheelbase l_b (cf. Figure 5.5) and an anticipation time t_{ant} correlated to the actuators capacities.

One drawback of using Frenet reference frame is the dependency on a known reference path, i.e., if the vehicle follows a dynamic target then the target trajectory must be accurately known in advance.

5.2.2.2/ Auto-Lane Change (ALC) based on ELC

In our work, Elliptic Limit-Cycle (ELC) (defined in section 5.1.2 has been used as appropriate path to have safe, efficient and flexible obstacle avoidance behavior in general, or ALC behavior in roadway navigation. To do so, we adapted the initial technique proposed for obstacle avoidance in any cluttered environments (cf. section 5.1.2, [11, 214]) to be more appropriate for road-way navigation. In what follows, details on the performed adaptations for road-way-navigation is given.

It was proposed [1] to adapt ELCs to the road-way use-case for lane change maneuvers by taking into account vehicle speeds and traffic rules in the dimensioning of the ELC trajectories of the controlled autonomous vehicle (cf. Figure 5.9).

To differentiate the multiple parts of the lane change maneuver, a convention is used based on Worrall and Bullen definition [227] in: **head portion, lane change part, tail portion** (cf. Figure 5.9). The ellipse of influence has been expanded laterally and longitudinally based on the traffic rule regulation and the vehicle dimensions. Indeed, based on the French traffic regulation, a minimum lateral distance of $L_{distance} = 1.5m$ has to be left during the tail portion and a longitudinal safety distance equivalent to the distance

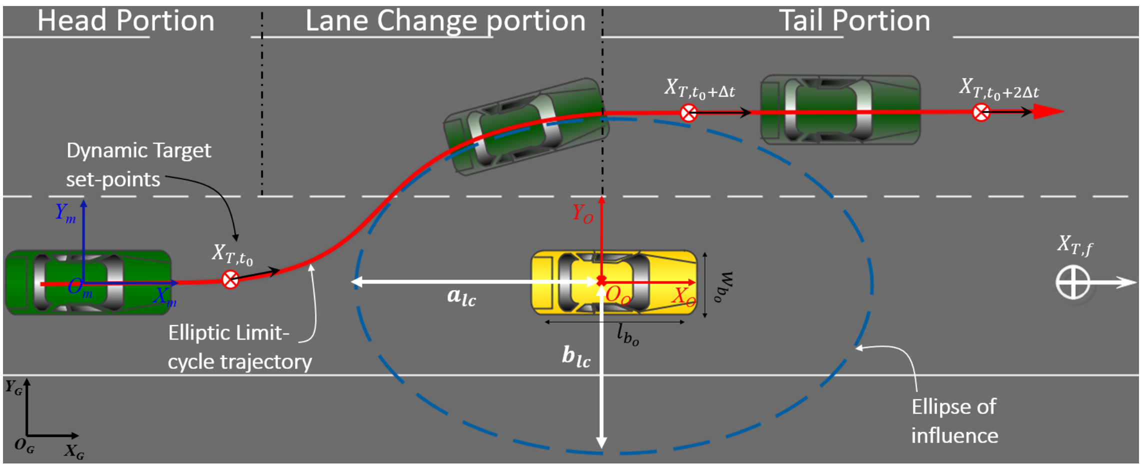


Figure 5.9: Lane Change maneuver: Overtaking yellow vehicle with ALC and LKA

traveled over $t_s = 2\ secs$. The parameters of the ellipse are then given by the following equations:

$$\begin{cases} a_{lc} & = & 0.5\ l_{b_o} + t_s\ v_r \\ b_{lc} & = & w_{b_o} + L_{distance} \end{cases} \tag{5.15}$$

with l_{bo} the wheelbase of the vehicle, w_{bo} the vehicle track (cf. Figure 5.9 and v_r is the relative velocity between the ego-vehicle and the vehicle to overtake (which is characterized as an obstacle and will be named ahead-vehicle). In this work, we assume full knowledge of the surrounding vehicles information. The orientation of the ellipse of influence follow the orientation of the corresponding overtaken vehicle.
Making use of the relative velocity makes our defined ellipse of influence variable in size in such a way that it adapts to relatively small changes in the dynamic of both concerned vehicles (cf. section 5.2.3 and Figure 5.12). In this way, if the overtaken vehicle is slowly braking or accelerating for example during a lane change, the ellipse will adapt longitudinal to be smaller or larger respectively.

As for the navigation, for each time step the ELC is calculated while taking as initial parameters the current ego-vehicle configuration. Here, the set-point configurations are taken within the generated ELC trajectories. The extraction of these set-points is based on a heuristic [9] defined in section 5.1.2, where at each sample time the intersection between a circle surrounding the ego-vehicle (defined by a radius R_s and a center corresponding to the origin of the reference frame linked to the ego-vehicle) and the pre-planned ELC is calculated (cf. Figure 5.10). The target set-points are given by:

- The intersection point (x_T, y_T) corresponding to the position of the dynamic set-point.
- The orientation θ_T corresponding to the tangent to the ELC w.r.t. $(x, y)_G$ at the intersection point (x_T, y_T).
- the velocity v_T corresponding to the velocity of the ego-vehicle so that the dynamic target set-point is always ahead at an Euclidean distance offset R_s.

R_s in the other side is fixed in the same why as R_{curv} empirically while taking into account

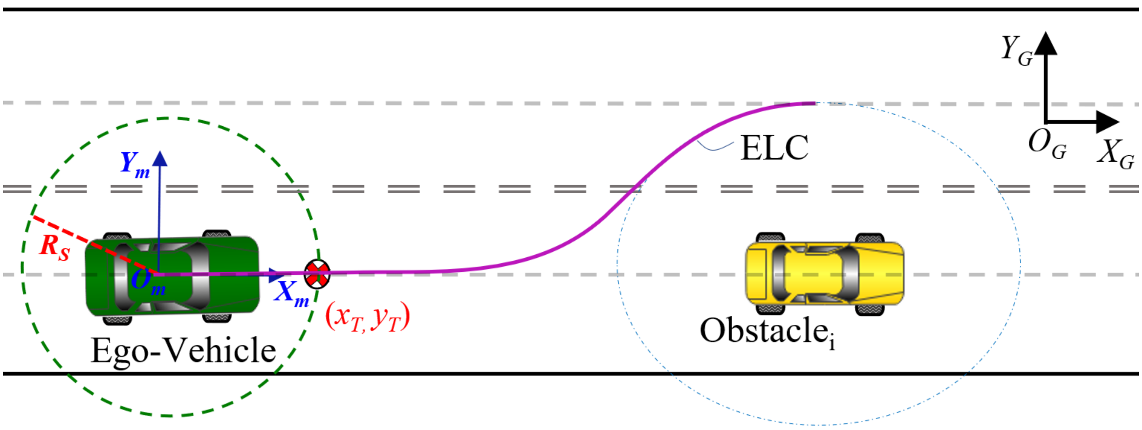


Figure 5.10: Homogeneous set-points definition based on dynamic target tracking: ALC based ELC

the velocity of the vehicle, the length of the wheelbase and an anticipation time t_{ant} correlated to the actuators capacities. A possible improvement of selection R_s et R_{curv} can be including the curvature of the trajectory into the computations which will allow to take tight turn. Nevertheless, in all the use-cases addressed in our work, the introduction of this curvature is not essential.

5.2.2.3/ Criterion for a complete lane change maneuver

Once the target selection performed, we shall define a criterion to specify when the ahead-vehicle is completely avoided. A short term goal or destination referred to in this work as the "*Final Target* $X_{T,F}$", is defined in the route planning (block 1 in Figure 5.6) within the road network. It is used for the definition of a criterion to specify that the vehicle is completely overtaken. Figure 5.9 shows the positioning of $X_{T,F}$ during a lane change maneuver.

A specific reference frame (shown in Figure 5.9) is then defined linked to the ahead-vehicle, that has the following characteristics (cf. Figure 5.9):

- X_O axis connecting the center of the ahead-vehicle (x_{obs}, y_{obs}) to the center of the target $X_{T,F}$. This axis is oriented towards this target.
- Y_O axis perpendicular to the X_O axis.

This reference frame is applied in order to change the position of the ego-vehicle $(x, y)_G$ given in the global reference frame towards the reference frame linked to the ahead-vehicle $(x, y)_O$. This is performed while using the following homogeneous transformation:

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_O = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & x_{\text{obst}} \\ \sin\alpha & \cos\alpha & 0 & y_{\text{obst}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_G \tag{5.16}$$

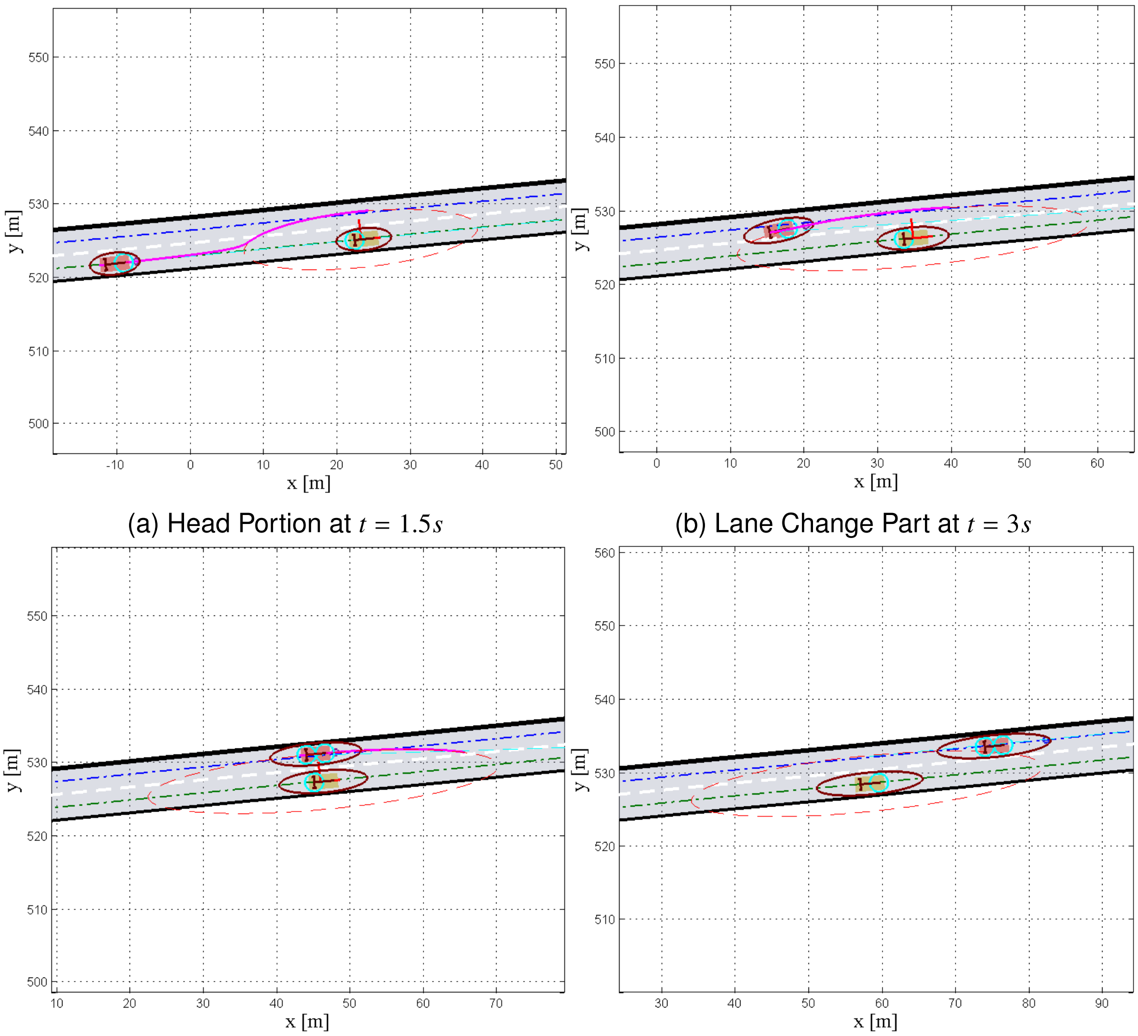
with α the rotation angle between the global reference frame and the defined ahead-vehicle reference frame [11].

In this way, if the position of the ego-vehicle along the X_O axes of the obstacle reference frame is greater or equal to 0 ($x_O \geq 0$) and the lateral distance $D_{lat_{ego}}$ of the ego-vehicle to the targeted lane is smaller then a defined minimal distance $D_{lat_{min}}$, the overtaking is considered completed. This step is important in the decision-making process and validation as we will see it in Chapter 6.

5.2.3/ Demonstrative simulations of the elementary controllers

The simulation results are based on experiments performed on a Matlab car simulator that has been implemented to test the developed algorithms (cf. Chapter 7).

To demonstrate the efficiency of the detailed architecture an exemplary overtaking maneuver scenario is shown in Figure 5.11. The ego-vehicle is traveling on the right part of a two-lane highway. The ahead-vehicle (which maintains a constant velocity) is considered as an obstacle by the ego-vehicle. A complete lane change maneuver from the head portion to the tale portion starts and all the elements described in previous sections are generated and are shown in Figure 5.11.



(a) Head Portion at $t = 1.5s$

(b) Lane Change Part at $t = 3s$

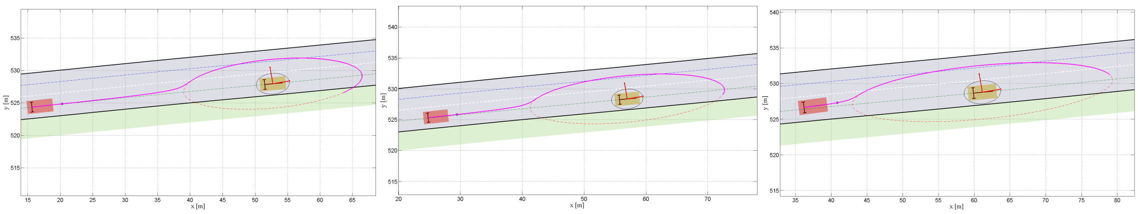
(c) Lane Change Part at $t = 4.5s$

(d) Tail portion at $t = 6s$

Figure 5.11: Simulation results of an overtaking maneuver.

Figure 5.12 shows the adaptability of an Ellipse of Influence (EI) to relatively small changes in the dynamic of both concerned vehicles. In this way, if the overtaken vehicle is slowly braking or accelerating for example during a lane change, the ellipse will adapt longitudinal to be larger or smaller respectively.

A navigation scenario illustrating the elements described in previous sections is shown in the following simulation video https://shorturl.at/orKY3 where a Lane Keeping Assist (LKA) is followed by a complete lane change maneuver from the head portion to the tale portion. Further simulations of these controllers and their interactions are shown in chapter 6.



(a) EI at $t = 1.5s$ (b) EI at $t = 2s$ (c) EI at $t = 2.5s$

Figure 5.12: The Ellipse of Influence (EI) adaptability during a lane change maneuver according to the dynamic changes of the ahead-vehicle

5.3/ Proposed safety management strategy

5.3.1/ Problem statement

The state of the art review conducted in chapter 2 led to the conclusion that an important challenge in the field of AV risk assessment is to find the perfect balance between autonomous navigation criteria like : smooth navigation, guarantee comfort of the passenger, high level of safety insurance, and imposed constraints such as: uncertainties, complexity, respect traffic laws and not being too conservative in the navigation. In this work, we define a risk management strategy based on two stages which are:

- The risk assessment strategy (cf. Section 5.3.2): Used in order to analyze the actual driving situation and predict potential collisions in the purpose of choosing the most suitable maneuver regarding the actual driving scene while taking into consideration any constraints or traffic condition that are known at the time of planning.
- The safety verification mechanism (cf. Section 5.3.3): Since every traffic situation is unique, it is necessary that the decided/planned maneuvers be always verified during navigation of the vehicle. Criteria are defined in this purpose in order to validate the first step assessment and alert us in case of any anomaly in the prior configuration of the environment in order to ensure even more AV safety in uncertain environment and changing dynamic/behaviors of the surrounding vehicles.

In what follows, details about the above stages and the corresponding safety criteria are given.

5.3.2/ Safety assessment based on Extended Time-To-Collision (ETTC)

The state of the art review conducted in chapter 3 detailed the use of deterministic risk indicator (cf. Table 3.2) in the literature. The review led to the conclusion that despite the good number of metrics that exist in the literature, the Time-To-Collision (TTC) and its derivatives remains the most used risk metric. However, this metric suffers from the lack of context and singularities may occur in some configuration and the TTC alone is insufficient as a risk indicator for managing complex situations. In addition, the common definition of the TTC is restricted for a specific path to detect longitudinal collision and for well-defined scenarios such as car following.

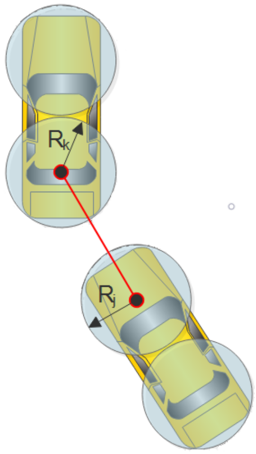


Figure 5.13: ETTC between ego-vehicle and surrounding Obstacle-Vehicles

To overcome this issue, extended definitions of the TTC have been proposed in multiple works [102, 118, 153, 223] and addresses the problem from a planar perspective where vehicles are considered in a two-dimensional plane. The method used in [102] relies on a straightforward way to determine when the ego-vehicle touches another obstacle-vehicle present on the scene. The authors propose a collision algorithm based on circles surrounding the vehicles. When the distance between the centers of two circles is equal to the sum of the two radii, the ego-vehicle has touched the obstacle-vehicle. To improve the formulation given in [102], a two-circles model ([1], cf. Figure 5.13) to surround each vehicle in the environment is considered. In this way, the front and the rear collision risk detection are addressed because this configuration reflects in a better way the shape and the dimensions of vehicles. The front circle is centered at the center of the front axle and the rear circle is centered at the center of the rear axle. They are described with a common radius reflecting the dimension of the vehicle. Similarly to the previous work, when the distance between the centers of two circles is equal to the sum of the two radii, the ego-vehicle has touched the obstacle-vehicle (for simplification they will be named k and j respectively). For both collision detection, the two-circles algorithm selects the pair of ego-obstacle circles by measuring the distances between the front and the rear circles. The closest ones with the smallest distance are considered at each time step (cf. Simulation figure 5.21 and simulation video - https://shorturl.at/orKY3).
The extended TTC (first proposed in [102]) is then derived given the radius, position, velocity and acceleration for every vehicle, following equation (5.17):

$$\begin{aligned}&\left[x_k + v_{xk}ETTC_{kj} + \tfrac{1}{2}a_{xk}ETTC_{kj}^2 - \left(x_j + v_{xj}ETTC_{kj} + \tfrac{1}{2}a_{xj}ETTC_{kj}^2\right)\right]^2 + \\ &\left[y_k + v_{yk}ETTC_{kj} + \tfrac{1}{2}a_{yk}ETTC_{kj}^2 - \left(y_j + v_{yj}ETTC_{kj} + \tfrac{1}{2}a_{yj}ETTC_{kj}^2\right)\right]^2 \\ &= (R_k + R_j)^2\end{aligned} \tag{5.17}$$

Where $ETTC_{kj}$ is the unknown extended TTC value between the vehicle j and the vehicle k, R_j and R_k (x_j, y_j) are the radius of each vehicle and (x_k, y_k) are the vehicle's coordinates, (v_{x_j}, v_{y_j}) and (v_{x_k}, v_{y_k}) are the speed components on X and Y direction, (a_{x_j}, a_{y_j}) and (a_{x_k}, a_{y_k}) are the acceleration components on X and Y direction. The smallest root value of this quartic equation is the ETTC value. A report analysis have been performed in [27] to study the limits of the TTC extension. We came at the conclusion that calculating ETTC in cases where the velocities are constant on parallel paths does not have an added value, the calculation of the classical TTC is sufficient. This metric is on the other hand very useful if the velocities vary or if the paths cross. Here again it was noted that the vehicles must be on the point of intersection at the same time otherwise the roots of the polynomial function contain an imaginary part. This result can be exploitable in the sense that if the roots contain an imaginary part, it means that the vehicles are not intended to collide with the detected input parameters.

The output ETTC will be directly used as input to our decision-making strategy (cf. Chapter 6) allowing us to make a suitable maneuver decision while taking into consideration any constraints or traffic condition that are known at the time of planning. However, we must be able to automatically abort the decided maneuver in case of any unexpected approaching objects such as other objects and road users entering the planned course of the vehicle. The vehicle must then be able to re-plan by determining an alternate route, i.e., the emergency trajectory or the evasive maneuver, which the car will pursue instantly to avert an accident and guarantee safety all the time. In this case, extensive testing to simulate all possible behaviors of other traffic participants can be used however it is a time consuming and an unworthy task to realize considering the uniqueness of each traffic situation, and can only prove that a system is unsafe, but is not able to propose an alternative. In this situation, it is necessary that the decided/planned maneuvers be always verified during navigation of the vehicle. This has been called in the literature online safety verification [14] or formal verification and answers to this challenge. It has been used in many works of the literature [14, 155] and are reviewed in Chapter 3, section 3.7. This task is performed in this work through a second threat measure dedicated to the safety verification mechanism and will be detailed in section 5.3.3. It is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver in a retrospective manner. The choice have been made in separating risk assessment and the safety verification in the objective of being consistent avoiding unnecessary switch in the plan. This reasoning will be discussed and detailed in the decision making strategy in Chapter 6.

5.3.3/ Safety verification based on inter-vehicular distance prediction

The safety verification is applied in real-time, during the maneuver achievement, where a safety verification mechanism is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver. This is performed through a safety criterion-based on the study of the dynamic of progression of the inter-distance between vehicles [1] that is used in order to estimate the maneuvers risks during the whole navigation task. The assumption considered in the definition of this criterion is that if nothing changes in the initial expected dynamic of all the vehicles in the environment including the ego-vehicle, the predicted evolution of the inter-distance between vehicles is not supposed to change. The following sections will detail the used criterion.

5.3.3.1/ Static Predicted Inter-Distance Profile (S-PIDP)

To illustrate the reasoning around this criterion, we studied a predefined Static Predicted Inter-Distance Profile (S-PIDP) built off-line during normal conditions (proposed in [1]) specifically during a lane change maneuver while following the definition on a complete lane change maneuver that divides the lane change maneuver in three portions: head

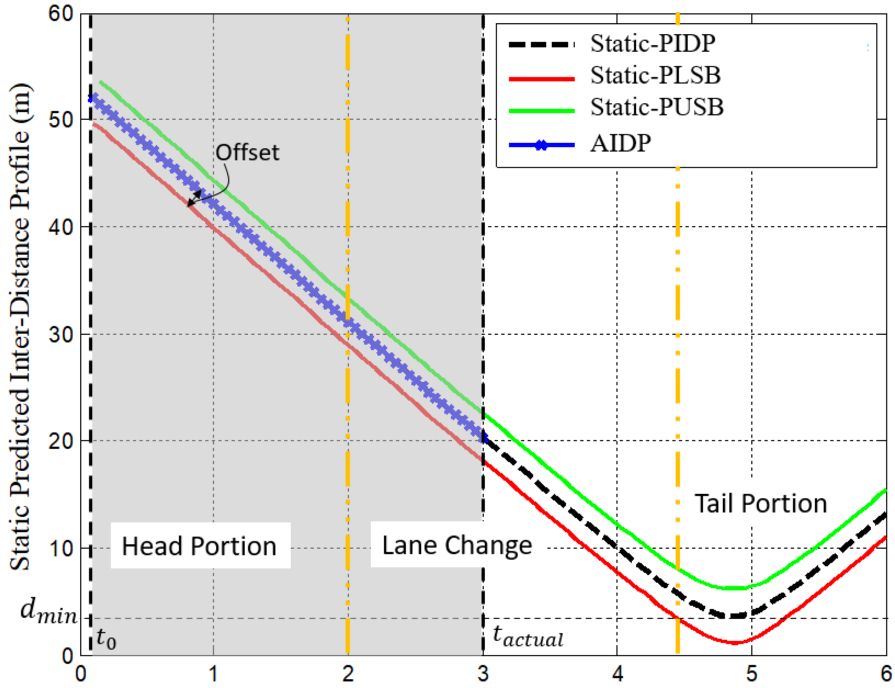


Figure 5.14: Static Predicted Inter-distance Profile (S-PIDP)

portion, lane change part, tail portion (cf. section 5.2.2.2 and Figure 5.9). The S-PIDP is constructed as the Euclidean distance between the ego vehicle and the candidate obstacle-vehicle for the overall lane change maneuver and is shown in Fig. 5.14. From the figure we can deduce two properties:

1. The first one is the shape of the profile, the distance decreases as we get close to the lane change part while respecting the dimensioning of the ellipse of influence and then increases during the tail portion.

2. The second is the minimal distance d_{min} (cf. Figure 5.14) during a lane change maneuver that generally corresponds to the moment where the ego-vehicle is in the adjacent lane and the vehicles are side by side. This latter property, is satisfied by our dimensioning of the ellipse of influence during lane change or evasion to the emergency lane.

A Static Predicted Lower Safety Boundary (S-PLSB) is allowed, that is fixed while taking into account the vehicles' uncertainties over their mutual position and relative velocities. The risk of collision increases when the progress of the Actual Inter-Distance Profile (AIDP) between the vehicles is not conform to the expected one.

The following lateral errors are then calculated as anomaly detection criteria for S-PIDP to assess the dangerousness of the situation (cf. Figure 5.15) that will allow us to detect any anomaly in the evolution of the distance as the following:

- Err_1 is the difference between the S-PIDP and the AIDP calculated for each time step: $Err_1(n) = \widehat{AIDP} - SPIDP(n)$
- Err_2 is the difference between the AIDP and the S-PLSB computed for each time step: $Err_2(n) = \widehat{AIDP} - SPLSB(n)$

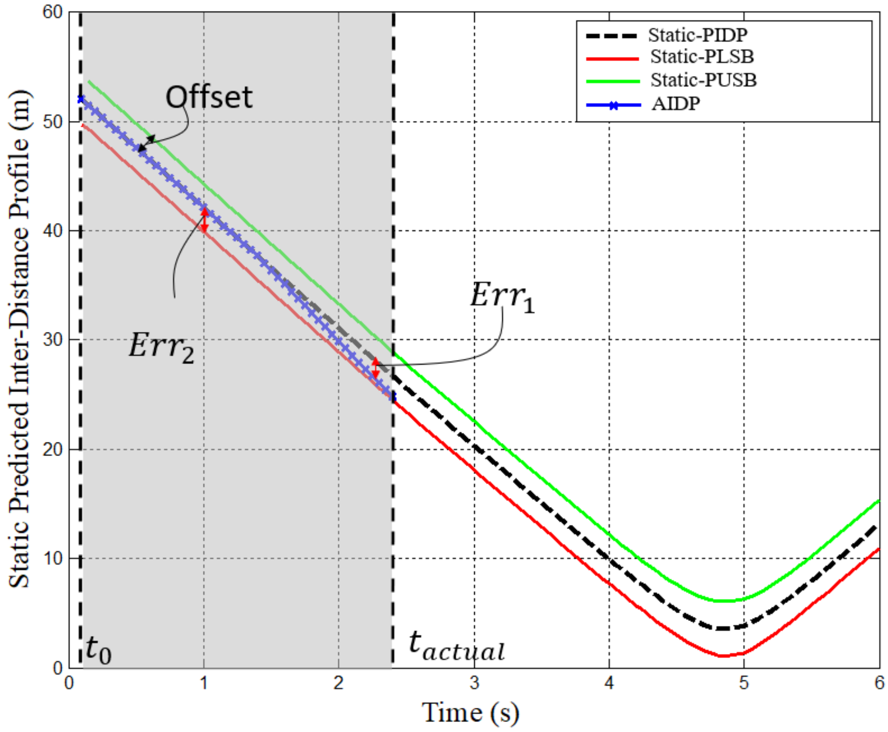


Figure 5.15: The lateral error definition of the S-PIDP between ego-vehicle and surrounding obstacle-vehicle

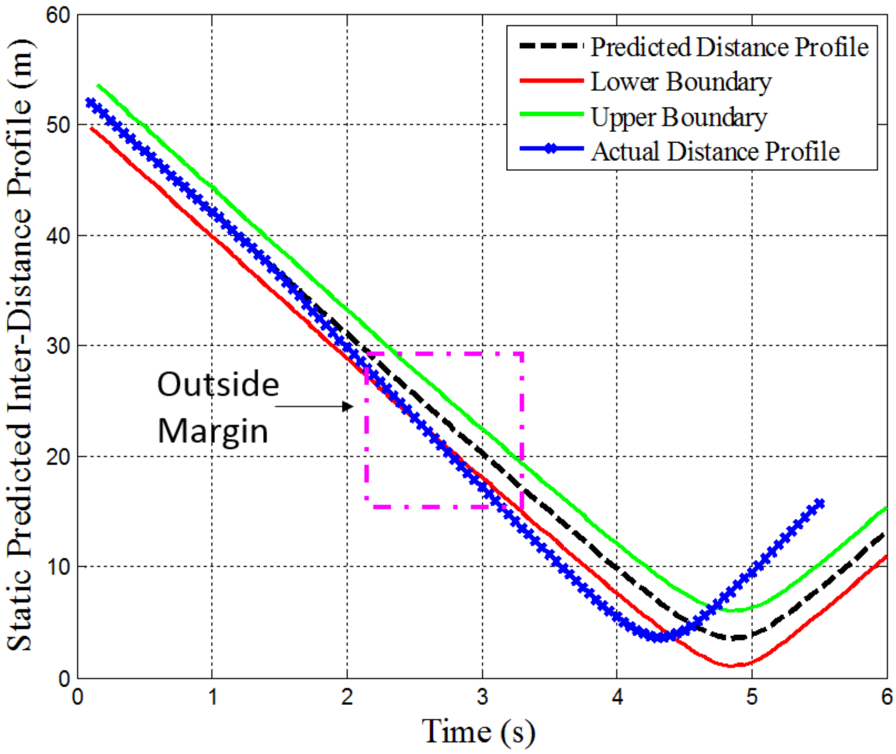


Figure 5.16: S-PIDP during a change in the dynamic of progression of the ahead-vehicle

The S-PIDP performed well in case of confirmed anomaly however in simulation where slight changes in the surrounding vehicles dynamic, the S-PIDP detects an anomaly where in reality there is not. As an example, Figure 5.16 shows a simulation where a change in the ahead-obstacle velocity is explicitly injected in the system at $t = 1.5s$ from $V_{O1} = 12m/s$ to $V_{O1} = 10m/s$ during the lane change maneuver. The AIDP goes outside the lowest safety margin, which will result clearly to a false alarm.

5.3.3.2/ The Dynamic Predicted Inter-Distance Profile (D-PIDP)

To palliate to the previously stated issue, we proposed in [2] to account for this dynamic changes and interactions between vehicles by introducing the use of predicted trajectories of road users over a time prediction horizon in the calculations of the inter-distance and while using the same concepts presented above. This is performed through an updated criterion called the Dynamic Predicted Inter-Distance Profile (D-PIDP) detailed in the following section.

Predictions of the vehicles' trajectories In order to account for the dynamic aspects of updating and reconfiguration for this safety-criterion, let's first introduce our trajectory prediction strategy. To better understand the approach, we will analyze the system during the lane change maneuver, but the same reasoning is applied during the whole navigation.

Predictions of the vehicles' trajectories is performed while using a *"maneuver-based formalism"* (cf. Chapter 2, section 3.2.2) where the predicted trajectory is adapted to the on going maneuver over a longer prediction horizon. Prediction of the ego-vehicle pose during the lane change is performed according to the solution of the Elliptic Limit Cycles (ELC) trajectories differential equation defined in equation (5.1) and equation (5.2) (cf. section 5.1.2). The latter gives the pre-planned trajectory, where the ego-vehicle's position is considered at each sample time as initial configuration of these ELC trajectories. This definition is particularly important in this work as the set of points constituting the pre-planned ELC trajectory will be considered as a prediction for future vehicle motion. In this work, it is assumed a constant velocity profile in the predictions.

An overall lane change trajectory $\mathbf{x}_{\text{pred}}(t) = (x(t), y(t))$ is then constructed using the elliptic limit-cycle set-points representing the head and the lane change part (cf. section 5.2.2.2). The intersection point between the pre-planned elliptic limit-cycle trajectory and the left lane corresponds to the starting of the tail portion. A temporal tail portion is then calcu-

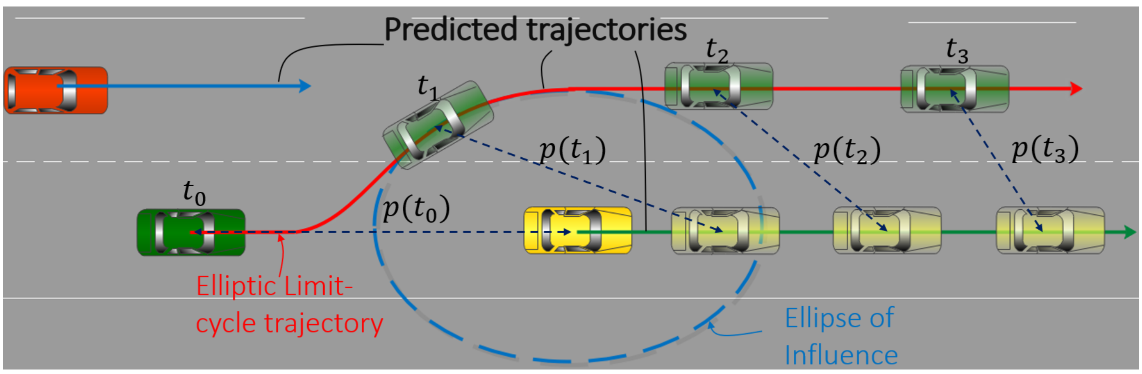


Figure 5.17: Predicted Trajectories during lane change maneuver

lated and combined to constitute the overall lane change trajectory (cf. Figure 5.17).
Numerous study have been performed in order to assess the duration ranges of a lane [137, 164]. In this work, the time prediction horizon $T_{pred}[s]$ corresponds to an estimation of the required time for the vehicle, given a constant velocity to travel the curvilinear distance of the overall trajectory change.
On the other hand, we suppose that the obstacle-vehicles follow a global path already defined to be the center-line of the lane at a constant velocity and the prediction trajectories $\mathbf{x}_{\text{pred},obs}(t) = (x_{obs}(t), y_{obs}(t))$ during the lane change are constructed for $T_{pred}[s]$ based on their expected behaviors.

The definition of the Dynamic Predicted Inter-Distance Profile (D-PIDP) Once the predictions of all vehicles is performed, the following step are followed:

- The DPIDP is calculated as the Euclidean distance between the consecutive points of the predicted state vector of the ego-vehicle and the predicted state vector of the chosen obstacle-vehicle for each time step of the prediction as shown in Figure 5.17 through $(p(t_0), p(t_1), p(t_2), p(t_3)$. The resulting overall curve $p(t)$ defined over T_{pred} is shown in Figure 5.18 and has the properties 1. and 2. defined in subsection 5.3.3.1.
- A control time horizon T_{ch} is then defined as the time to update $p(t)$ and is chosen to

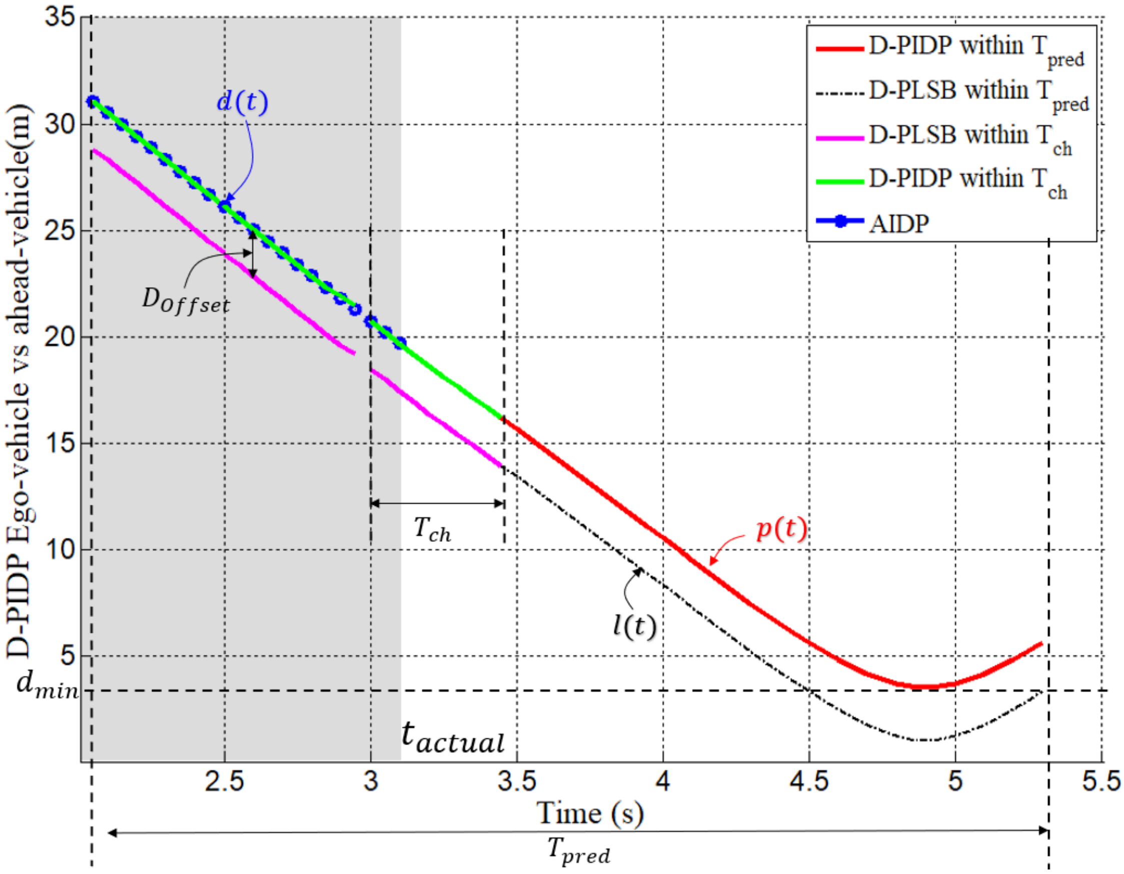


Figure 5.18: Definition of the D-PIDP between ego-vehicle and a surrounding Obstacle-Vehicle

be greater then the the sampling period T_s and smaller then time prediction horizon T_{pred}. For each T_{ch}, $p(t)$ will be re-evaluated between the predicted state vector of the ego-vehicle and the predicted state vector of the chosen obstacle-vehicle (cf. Figure 5.17). The risk of collision increases when the progress of the Actual Inter-Distance Profile (AIDP), defined as the curve $d(t)$ in Figure 5.18, get close the .

- Similarly to the Static Predicted Inter-Distance Profile (S-PIDP), a Dynamic Predicted Lower Safety Boundary (D-PLSB) is constructed as the projection (parallel curve) of $p(t)$ with an offset shift denoting a possible authorized degree of freedom over the difference between the actual displacement (distance, velocity) between the vehicles. This is shown in Figure 5.18 as the curve $l(t)$.

This way of defining the D-PIDP safety-criterion gives the system an average time (T_{ch}) to confirm or not the dangerousness (given by the anomaly criteria detailed in subsection 5.3.3.2) of the situation and to act accordingly. This way of reasoning under uncertainty will eventually help reduce false alarm and improve performance. This latter property is shown in a scenario, similar to the defined one for the S-PIDP in Figure 5.16 where a slight change in the ahead vehicle is explicitly injected during the lane change maneuver. In comparison to the S-PIDP, we can notice that the D-PIDP reconfigure within an average control horizon time T_{ch} and adapt to the changes as the safety is ensured (cf. Figure 5.19). The actual inter-distance didn't cross the lower safety boundary.

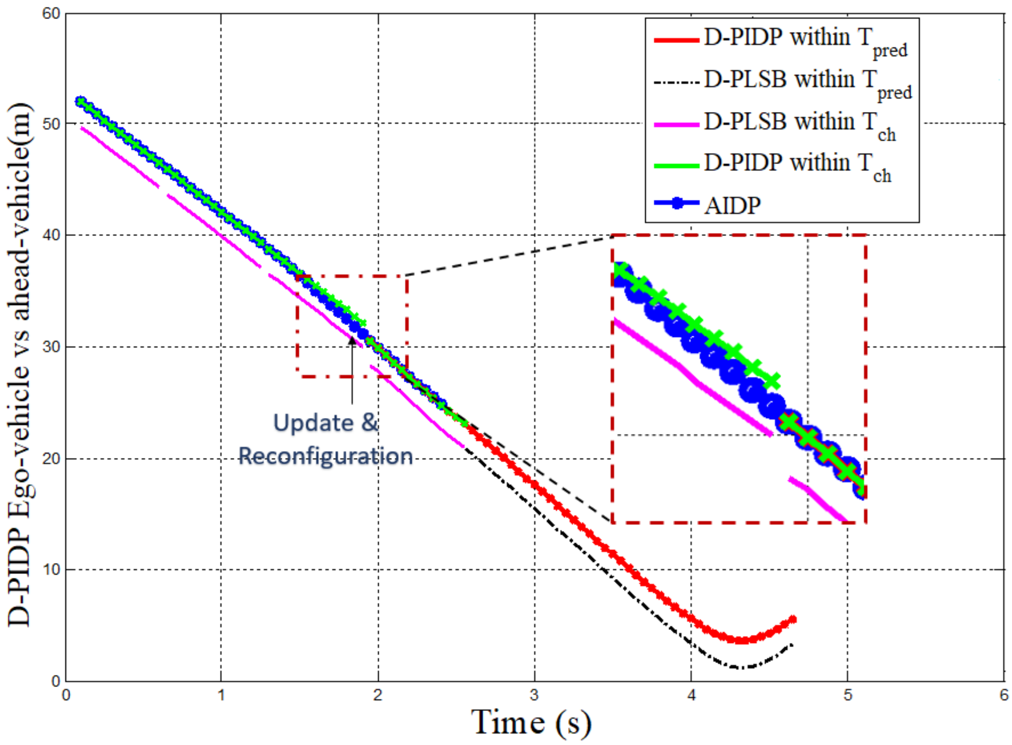


Figure 5.19: D-PIDP during a change in the dynamic of progression of the ahead-vehicle

In what follows, to assess the dangerousness of the situation, an improved formulation of the anomaly detection criteria defined in subsection 5.3.3.1 and figure 5.15 is proposed.

The D-PIDP anomaly detection criteria In [4] , we introduced an anomaly detection criterion called the *"critical time"* ($t_{critical}$) (cf. Figure 5.20). $t_{critical}$ is the time interval between the time t_ξ of the first variation of $d(t)$ and the time of the intersection point t_{inter} between $d(t)$ and $l(t)$. The first variation of the function $d(t)$ is characterized according to the time t_ξ where the error $\xi(t)$ between the derivative of $d(t)$ with respect to the derivative of the expected profile $p(t)$ at time t is greater then a small value ϵ i.e., $|\xi| > \epsilon$.
The critical time will be then:

$$t_{critical} = t_{inter} - t_{|\xi|>\epsilon} \tag{5.18}$$

This criterion compared to the lateral errors defined in subsection 5.3.3.1, combines two properties. The first one is that the AIDP crossed the lower boundary (through the calculation of the intersection point). This will allow us to detect the endangered obstacle-vehicles. The second one is the information on criticality of the situation, the smaller $t_{critical}$ is (due to a quicker deceleration for instance of the ahead obstacle-vehicle to overtake), the steeper the descent (cf. Figure 5.20). In what follows, some simulations are shown to

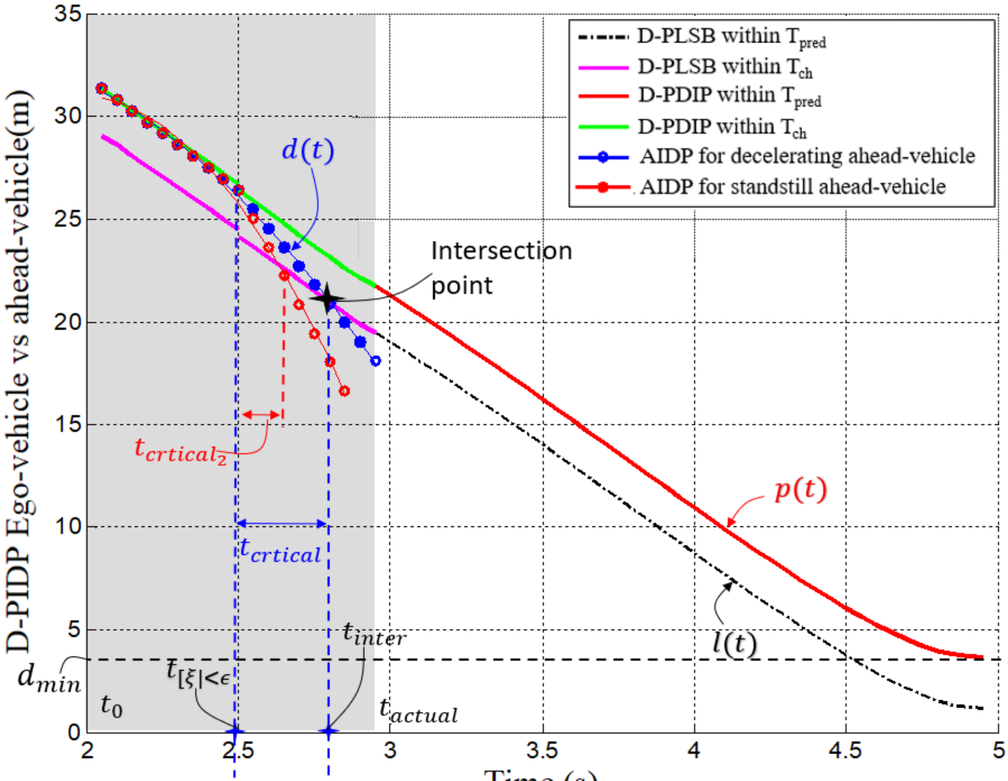


Figure 5.20: The anomaly definition for the D-PIDP between ego-vehicle and a surrounding Obstacle-Vehicle

illustrate the ability of the proposed metric to detect anomalies.

Evaluation of the D-PIDP anomaly criteria In this simulation, we will show the capability of the defined safety-criterion for anomaly detection. For the simulation shown below, it is considered what follows:

- The scene (cf. Figure 5.21) is constituted of three vehicles in a two-lane road: two

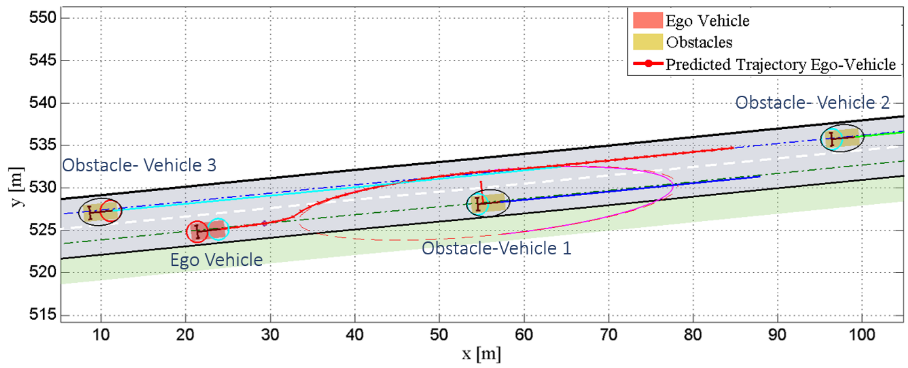


Figure 5.21: Car Simulator: Predicted Trajectories

vehicles on the right lane including the ego-vehicle (named respectively ego-vehicle and obstacle-vehicle 1 O_1) and one vehicle on the left lane obstacle-vehicle 2 O_2.

- $Vego_{max} = 23m/s$, $V_{O1} = 12m/s$, $V_{O2} = 25m/s$

In a nominal use-case, for the given scene (cf. Figure 5.21), the resulting DPIDP for each ego-vehicle/obstacle-vehicle pair is shown in Figure 5.22. In this case, the AIDP profile for each vehicle pair is conform to the expected D-PIDP.

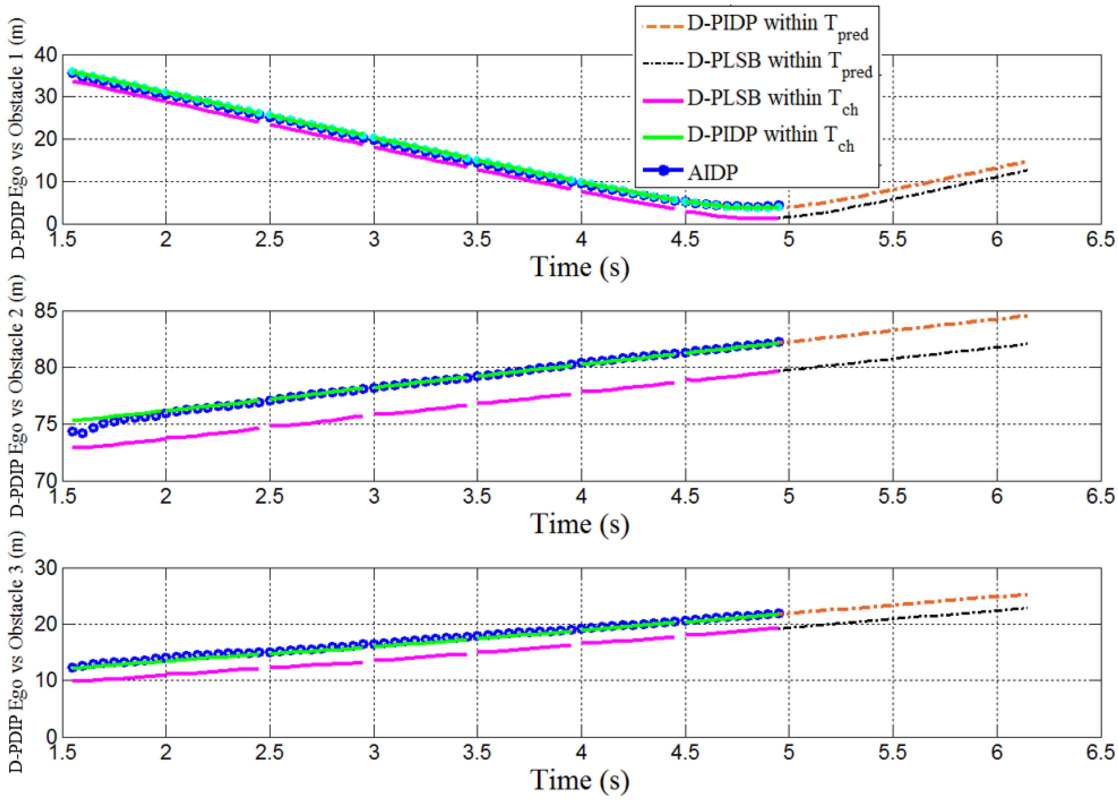


Figure 5.22: D-PIDP between ego-vehicle and surrounding Obstacle-Vehicles

In an emergency situation use-case, where for example the obstacle-vehicle in front suddenly brake and comes to stand still during a lane change maneuver and obstacle-vehicle

3 coming from behind in the left lane accelerating (cf. Figure 5.23) the following profiles shown in Figure 5.24 are generated.

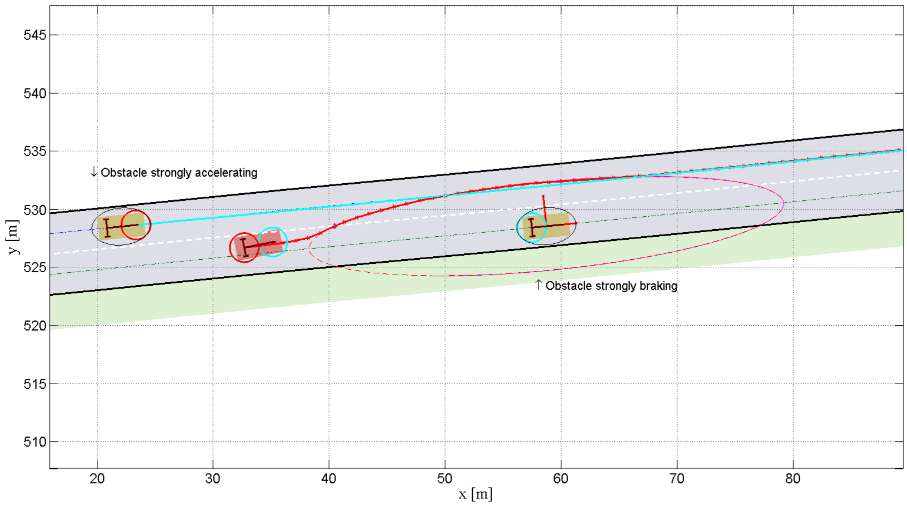


Figure 5.23: Example of a navigation in emergency situations (cf. Simulation Video - https://shorturl.at/qstLX)

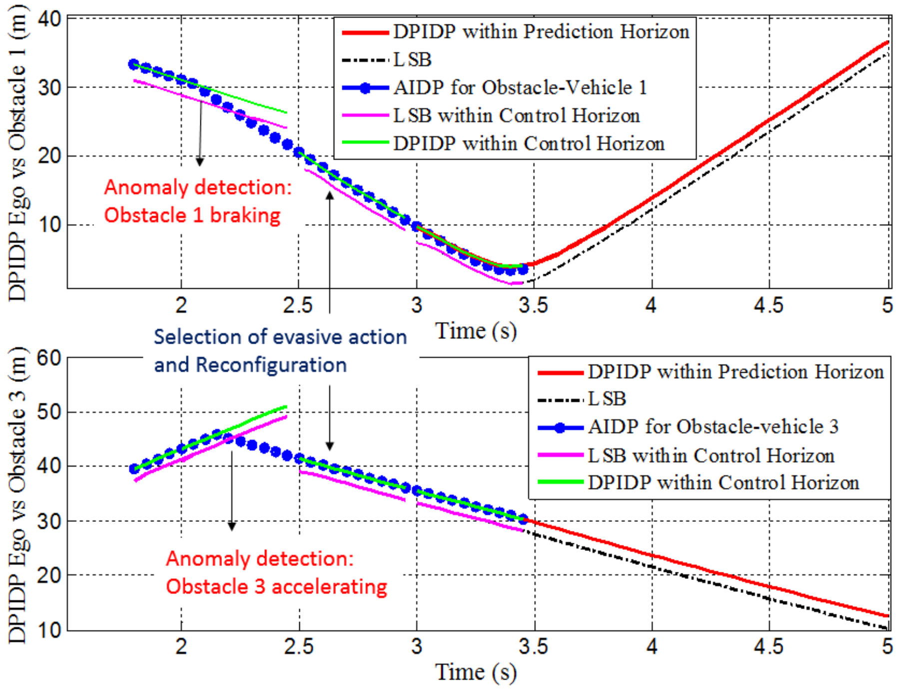


Figure 5.24: D-PIDP for the two obstacles

At the beginning of the simulation, the obstacle-vehicle 3 is far and slow enough to allow the lane change maneuver to start but suddenly accelerates. For the two vehicles, the danger threshold is reached when the AIDP crosses the defined PLSB generating an alarm given a positive $t_{critical}$ (cf. equation $^{(5.18)}$ and Figure 5.20). The vehicle configuration is shown in Figure 5.23 and the D-PIDP of the two obstacles is detailed in Figure 5.24.

At this point, a decision-making strategy that include an evasive plan is mandatory to handle this kind of situations. For this purpose, chapter 6 details the decision-making framework.

5.4/ CONCLUSION

In this chapter, it is presented the design of a Probabilistic Multi-Controller Architecture (P-MCA) for safe automated driving under uncertainties. In addition to the details given about the main modules (and their interactions) composing the proposed P-MCA, we justified the design choices of our architecture. Indeed, the P-MCA effectively links model-based approaches and Artificial Intelligence (AI) developments for intelligent vehicles navigation. This architecture assembles several interconnected modules that are responsible for: assessing the overall surrounding environment by evaluating the collision risk with all observed vehicles considering predictions of road user trajectories [1]; planning driving maneuvers [2]; making the decision on the most suitable actions; and investigating the possibility to decide evasive actions if required (cf. chapter 6, [4]).

In the first part of this chapter, the path planning through the used elementary controllers is presented and is highlighted an adaptation of previous work [214, 216, 217] in order to cope with road-way navigation constraints. Indeed, the proposed Probabilistic Multi-Controller Architecture (P-MCA) is the immediate extension of an already developed Multi-Controller Architecture (MCA) for the navigation of mono- and multi-vehicle navigation in cluttered environments. The proposed controllers are: the Adaptive Cruise Control (ACC), the Lane Keeping Assist (LKA) and the Automatic Lane Changing (ALC). Each vehicle's controller is constituted by a dedicated homogeneous dynamic set-points defined by a pose (x_T, y_T, θ_T) and a velocity v_T. It is highlighted that this set-point formulation is generic enough to define an important number of the vehicle's behaviors. An emphasize is made about the fact that the navigation is performed while tracking these set-points and not in a trajectory following behavior. At each sample time, one of the mentioned controllers is activated and the dynamic set-points are generated. The LKA and ACC controllers uses Frenet reference frame for the dynamic target set-points extraction (cf. subsection 5.2.2.1). The ALC on the other hand is based on Elliptic Limit-Cycle trajectory adapted to road-way navigation and it uses a heuristic to extract its set-points (cf. subsection 5.2.2.2). Once the set-points are defined, a unique stable control law is defined in section 5.1.5 shared by the defined controller will be used to stabilize the errors to zero and thus allow us to reach/track these assigned set-points.

In the second part, it is introduced the proposed approach for risk assessment. Our solution is based on a dual-safety criteria combining an Extended Time-To-Collision (ETTC) and a novel Dynamic Predicted Inter-Distance Profile (D-PIDP) between vehicles. The ETTC is used in the safety assessment strategy in order to analyze the actual driving situation and predict potential collisions. This metric will be later used in order to choose the most suitable maneuver while taking into consideration any constraints or traffic condition that are known at the time of planning.
The D-PIDP in the other side is used in the safety verification mechanism. Since every traffic situation is unique, it is necessary that the decided/planned maneuvers be always verified online during navigation of the vehicle. The D-PIDP is used for this purpose to define an appropriate analytical formulation of anomaly detection criteria called the critical time $t_{critical}$, allowing us to quantify the risks and the criticality of the driving situation beyond the time of planning especially in uncertain environment and changing

dynamic/behaviors of the surrounding vehicles.

Through this chapter several simulation results through a MATLAB car simulator have been shown to assess the good performance of the proposed modules.

In the next chapter, will be detailed the decision-making framework. The link between the decision-making and the defined risk assessment and verification criteria will be illustrated.

6

Sequential Decision Network for Maneuver Selection and Verification (SDN-MSV)

Contents

This chapter considers the problematic of decision-making for autonomous vehicles (cf. Chapter 4 for literature review on decision-making algorithms). The objective is to derive appropriate decision maneuver in nominal driving and guarantee safety in emergency situations. For this purpose, we propose to use a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) that utilizes multiple complementary threat measures to propose discrete actions and derive the appropriate maneuver in a given traffic situation. The SL-BDN allows us a safety retrospection and verification over the current maneuver risk and advise appropriate evasive action autonomously from moving obstacles.

6.1/ Problem Statement

Decision making for autonomous driving according to [194, 207] requires: ***rapidity*** in at least the planned decision for the driving maneuvers, ***coherency*** as the decision making module should be consistent in order to avoid any unnecessary switch in the plan, ***providentness*** meaning the module should foresee how the situation will evolve after some time or some maneuvers and include it in the decision making and ***predictability*** as the decision making should act while conforming to a human driver perception of safety and judgment for any situation.

In a traffic situation, the ego-vehicle has to adapt its behavior to the surrounding traffic environment. To pursue a lane change maneuver for example, the ego-vehicle has to position itself in a gap between traveling in the target lane while ensuring a safe distance between the nearby vehicle in its current lane during all the maneuver. Thus, for a lane change maneuver to be safe and feasible it should be planned such that we respect the followings:

- Avoid potential collision with other traffic participants.
- Respect traffic regulations.
- Satisfy the dynamic constraint and limitations of the vehicle.
- Respect the comfort of passengers with reduced level of lateral and longitudinal acceleration along with jerking.

This is performed in this work while using the Elementary controllers and the risk assessment strategy already defined in Section 5.2.2 and Section 5.3 respectively. In addition for a lane change maneuver to be safe and feasible [164], it should be desirable and advisable by the system in three conditions:

- Improve the driving conditions and maintain a desired velocity by overtaking a slow vehicle.
- Allow a rear faster vehicle to pass by changing lane to the right and let it pass in safety.
- Avoid any imminent collision.

The two first conditions will be defined in this dissertation as nominal driving situations. The third condition will be defined as emergency situations where the ongoing maneuver is aborted automatically and a re-planning is required by determining an alternate route, i.e., the emergency trajectory, which the car will pursue instantly to avert an accident and guarantee safety all the time. In all of these conditions, the decision making system should be able to predict and react to these changes and guarantee safety in all cases. Theses can be summarized in two scenarios illustrated in Figure 1.3 of Chapter 1.

In our work, this probabilistic decision-making strategy is defined as a part of the Probabilistic Multi-Controller Architecture (P-MCA) proposed in the previous chapter and is modeled as a sequencing of decisions that a self driving car should take by the means of a robust Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV).

6.2/ The proposed decision-making framework

The flowchart presented in Figure 6.1 illustrates the different proposed decision/validation sequences, and overall interactions between: the sequencing of decisions, the input risk assessment, the overall safety verification mechanism for all the obstacles present in the environment and the evasive strategy. This flowchart represent a focus in the block 2 of the overall Probabilistic Multi-Controller Architecture (P-MCA) proposed in chapter 5 (cf. Figure 5.6). As a reminder, the inputs of this block are: the perception/localization parameters of the ego-vehicle and the perceived surrounding vehicles, the road network defined by the centerline of the lanes and the predicted trajectories based on block 3b defined in subsection 5.3.3.2. The output is the most suitable decision used to select the suitable controller (block 3 and 3a or block 4) and regarding this decision that generates the appropriate control sequence (v, γ) (block 5 or block 4).

In the shown flowchart, three decisions are illustrated:

1. the first decision is a part of the Maneuver Decision Level (MDL) (cf. Section 6.3.2) where for each $i^{th} \in \mathbb{N} > N$ iteration time, the choice of action regarding the most suitable maneuver is made (cf. Figure 6.1). The probabilistic decision process is based on the current situation assessment, using the Extended Time-To-Collision (ETTC) detailed previously in Section 5.3.2 and while taking measurement uncertainty into account. The possible output maneuvers are: Lane Change Left (LCL), Lane Change Right (LCR), Keep Lane with ACC (KL-ACC), Maintain Velocity with CC (MV). These possible maneuvers are directly linked to the possible behavior that can be performed by the elementary controllers defined in subsection 5.2.2. To perform the maneuvers LCL and LCR the Automatic Lane Changing (ALC) controller is used whereas the MV maneuver uses the Lane Keeping Assist (LKA) controller and the KL-ACC uses in addition to the LKA the Adaptive Cruise Control (ACC) controller.

2. The second decision is a part of the Safety Verification Decision Level (SVDL) (cf. Section 6.3.3) where for each time step T_s, while the maneuver execution starts, a safety-checking regarding the action chosen in the MDL and a verification of the coherence of the maneuver with the predicted pre-planned trajectory is performed through the Dynamic Predicted Inter-Distance Profile (D-PIDP) [2] (previously detailed in Section 5.3.3). This decision level is used to detect and compensate for possible failure or unexpected behaviors and its possible outputs, which are: Maneuver is safe and Abort maneuver.

3. The third decision is a part of the Evasive Action Decision Level (EADL) (cf. section 6.4.2) where in case the verification procedure from the Safety Verification Decision Level (SVDL) advises to abort the maneuver, the system output the discrete evasive action based on the vehicles maximum capacities and on the endangered lanes i.e., the lanes where the anomaly is detected. The possible output are: Continue maneuver, Emergency braking, Emergency stopping lane (shoulder lane). Collision Mitigation is not considered in this work but can be included in the presented architecture given its modularity.

A most suitable decision is then obtained by maximizing a utility function over the possible alternatives of the action nodes (cf. Appendix A), given the available evidence.

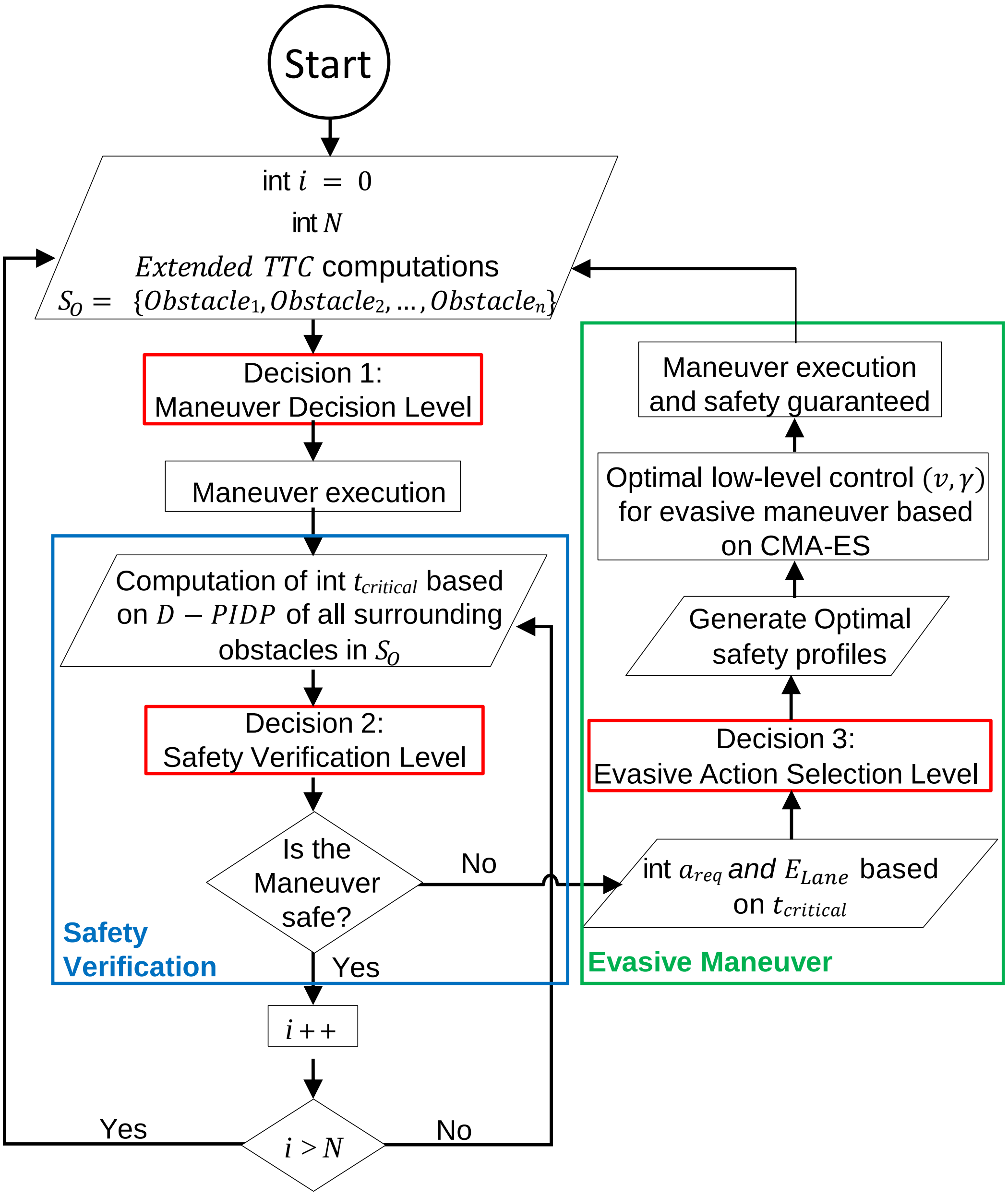


Figure 6.1: Flowchart illustrating the sequencing of decisions and safety verification for all surrounding obstacles. i is an integer value defining the iteration step. N is an integer value and is defined as $\lceil \frac{T_{ch}}{T_s} \rceil$ with T_s the sampling period and T_{ch} is the control observation horizon. S_O is the set of visible obstacles in the scene with memory tracking Id. $t_{critical}$ is the anomaly detection criteria defined in section 5.3.3.2. $ETTC$ is the Extended Time-To-Collision (ETTC). a_{req} is the required acceleration and E_{Lane} is the endangered lane (cf. section 6.4.2).

Then, based on the Evasive Action Decision Level (EADL) output of the Bayesian Decision network, it is proposed to compute the corresponding low-level control that follows the advised evasive maneuver (cf. Section 6.4). This is accomplished by defining optimal safety profiles that should be matched to stay safe based on a double threat metric: a Predicted Inter-Distance profile and a Predicted angular profile. The evasive strategy

is based on an evolutionary algorithm (EA), the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) that is able to reach a global optimum in few generations (cf. Appendix B). Indeed, this will allow the overall system to increase its degrees of freedom concerning the maneuverability of the vehicle and allow smooth changes during the evasion, thus ensuring the safety of the system and also the passengers' comfort. This methodology makes perfect sense when the evasive decision include a lane changing as the lateral constraints are the hardest to full-fill and a wrong swerve can be costly and fatal. In this work, we always favor the use or re-use of the already implemented modules by adapting its parameters when possible. This choice is performed when an emergency braking is required we activate in this case the ACC with the suited deceleration and set-points parameters. This fine-tuning is shown in detail in section 6.4.
Finally, a loop-back from the evasive maneuver is made towards the initialization of the algorithm once the safety state is reached to restart the decision-making process.

6.3/ Modeling interactions with decision Bayesian network

A Bayesian network (BN) is a directed acyclic graph representing a set of random variables and their conditional dependencies. Decision networks (DN) are an extension of BNs that allow us to support probabilistic reasoning, decision-making under uncertainty for a given system and yield the capacity to incorporate multiple decision criteria. A detailed description of the main properties of Bayesian Networks are given in Appendix A. In what follows we will detail the topology, the main components and the inference/decision-making calculations of the proposed DN.

6.3.1/ Topology of the Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV)

The topology of the proposed SDN-MSV is shown in Figure 6.2. The lanes are numbered from right to left by $i = 1..N_{lanes}$ with N_{lanes} number of the lanes of the road and $i = 1$ denoting the rightmost lane. In this work, for the sake of convenience, a two lane configuration to present this model is considered. However, this architecture is generic and can be extended to an N-Lane configuration. The proposed DN architecture is designed to handle numerous scenarios configuration, multiple decision criteria while taking uncertainty into account. The graphical representation of the Decision Network (DN) eases the connection between:

- The situation assessment level or the observation level (shown in figure 6.2 as the "current driving state of safety") that comes at the upper level of the SDN-MSV and is composed by chance nodes U_C. This nodes are defined using the proposed risk measures (cf. section 5.3) and different proprioceptive information.
- The decision-making level that comes at the lower level composed by utility nodes U_V used to define the cost related to the decision and decision nodes U_D to represent the choice of action among alternatives.

The nodes composing the SDN-MSV will be detailed through sections 6.3.2, section 6.3.3 and section 6.4.2.

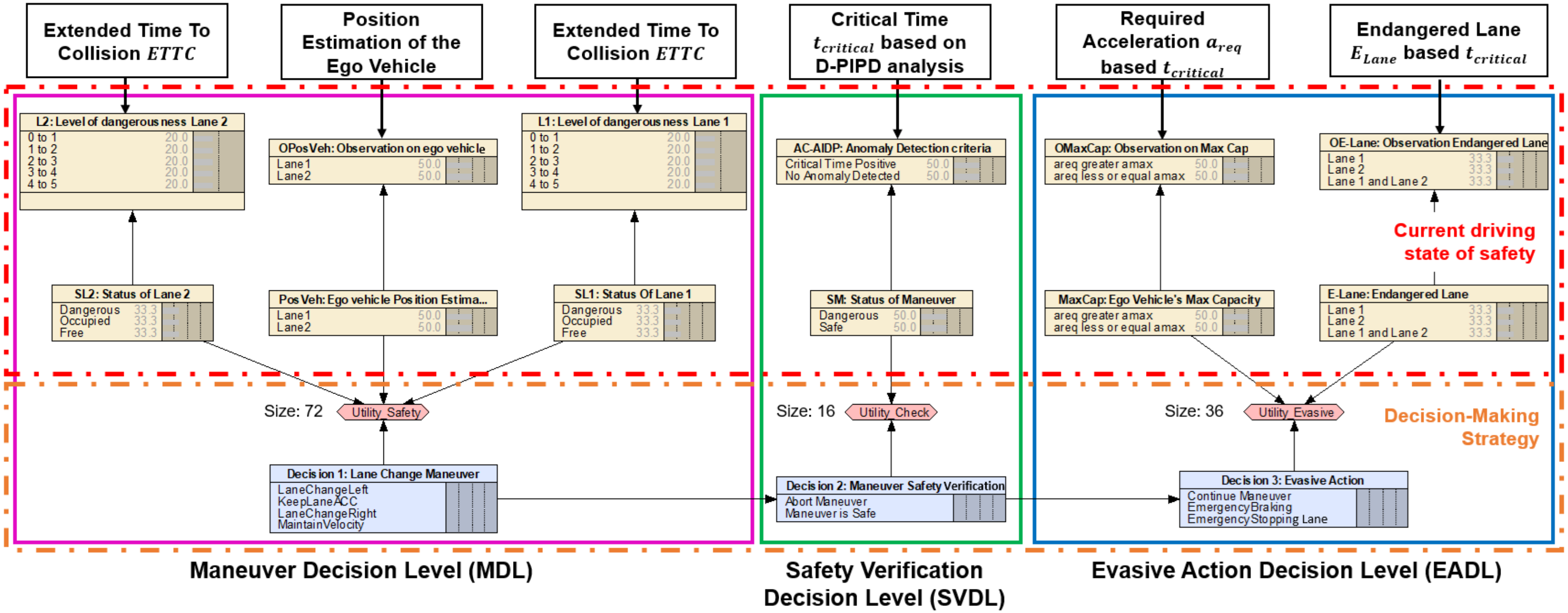


Figure 6.2: Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) nomenclature (developed while using Netica software)

The purpose of a multi-level DN instead of one is the ability to reason over a control observation horizon T_{ch} (cf. Figure 5.18). As the MDL is dynamic and the choice of action is instantaneously taken which means if a false alarm is triggered due to wrong information from the perceptive module for instance, the MDL will immediately abort the previous decision and compute another one. The SVDL, on the other hand allows us in this case to verify the coherence of the maneuver over the control time horizon T_{ch} and thus allows us a safety retrospection over the current performed maneuver. In addition, this way of modeling allows us a simpler comprehension of of the structure of the decision system. From a complexity perspectives, it allows to reduce the size of the Conditional Probability Table (CPT) of each node as well as Utility Tables (UT) size. This size is defined as follows [80, 163, 181]:

$$\text{size}(CPT \text{ or } UT) = s \cdot \left(s_p\right)^p \tag{6.1}$$

With s is the number of states, p is the number of parents and s_p is the number of parent states. The size of each utility table is shown in Figure 6.2.

6.3.2/ Maneuver Decision Level (MDL)

In the Maneuver Decision Level (MDL) the choice of action regarding the most suitable maneuver is made based on the current situation assessment, using the Extended Time-To-Collision (ETTC) detailed previously in section 5.3.2 and the position of the ego-vehicle in the lane. The MDL is an inspired version of the DN proposed by Schubert in [186] for lane change decision-making. Schubert used a deceleration to safety time (DST) as a threat measure to assess the danger of the navigation lanes status. However, the common definition of the DST is restricted for a specific path to detect longitudinal collision. The MDL is shown in Figure 6.3 and represent a micro version of the SDN-MSV shown in Figure 6.2, used to eases the understanding of the variables and their interactions.

	Type of node	**Description**
$\mathbf{SL_i}$	Continuous Node	Uncertain observation describing the level of danger of the lanes based on the ETTC.
$\mathbf{L_i}$	Discrete Node	The level of danger of the lanes based on the uncertain observation
PosVeh	Discrete Node	The vehicle position in the lane based on the uncertain observation
OPosVeh	Discrete Node	Uncertain observation denoting the estimated position of the vehicle in the lane

Table 6.1: The Maneuver Decision Level (MDL) variables

To derive the decision strategy for the most suitable maneuver to be achieved, the evaluation of the current state of the system is described by probabilistic variable, given in the DN by chance nodes. These nodes defining the structure of the MDL are described while emphasizing the different prior probabilities definition in the following sections. We will detail the probabilistic equations only in this section as for the other decision level the logic is the same.

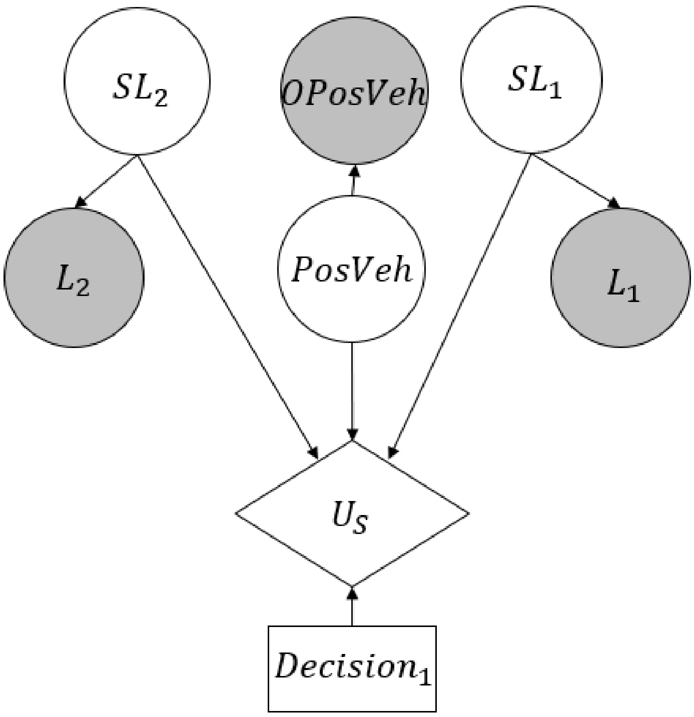


Figure 6.3: The Maneuver Decision Level (MDL) diagram

6.3.2.1/ SITUATION ASSESSMENT VARIABLES IN THE MDL

Observation on the vehicle's position ($OPosVeh$) The node $OPosVeh$ (cf. Figure 6.2) is a discrete node representing the uncertain observation denoting the estimated position of the vehicle in the lane. Lane information are estimated from OpenSteetMap (OSM) [120] for example. In a DN when a rough evidence on a random variable ($PosVeh$) is found, the concept of virtual evidence is used [163, 171]. A virtual evidence according to Pearl [171] corresponds to judgments based on undisclosed observation that are outside the network but have bearing on variables of the network. Such evidence is modeled as virtual node ($OPosVeh$), as a child of the random variable ($PosVeh$) affected by the observation. This virtual node ($OPosVeh$) will have a CPT that reflects the uncertainty of the observed variable ($PosVeh$).
The candidate lane is selected by checking the closest distance of the vehicle to the center-line of one of the lane based on the definition of the Frenet reference frame (cf. Figure 5.8). This estimation is augmented with uncertainty, as the situation assessment needs to account for erroneous estimation. In this sense, a prior knowledge of the sensing limitations and behavioral changes under all potential circumstances is mandatory. The authors in [29, 30, 31] proposed an interval analysis-based scheme to evolve several uncertainty sources such as: sensors latencies, harsh weather conditions or communication interference into the situational risk assessment.

This can also be modeled using an ellipse of uncertainty around the vehicle, propagated laterally and longitudinally in terms of: the distance traveled from position Pos_{k-1} to Pos_k and the vehicle's dimension. The overflow of this ellipse near the adjacent lane will be the uncertainty accounted in the ($OPosVeh$) node's CPT. To illustrate this principal, table 6.2

shows the confidence on the localization of the ego-vehicle. This is given by a probability that the ego-vehicle is in Lane 1 or Lane 2 at time t given the perception/localization information. This is read as: what is the probability of $PosVeh$ being in $Lane_1$ provided that $OPosVeh$ perceives that we are on $Lane_1$?

	OPosVeh	
PosVeh	*Lane 1*	*Lane 2*
Lane 1	0.98	0.02
Lane 2	0.02	0.98

Table 6.2: The conditional probability table for $OPosVeh$

Vehicle position estimation ($PosVeh$) The node $PosVeh$ is a discrete node denoting the vehicle position in the lane given the observation $OPosVeh$. The possible states are *Lane 1* or *Lane 2*. This node is initialized to be uniformly distributed as we have no prior knowledge on the state of the node.

Using Bayes theorem, the law of total probability and the chain rule (cf. Appendix A), the inference of $PosVeh$ given the available evidence $\mathbf{e}$ on $OPosVeh$ will be then:

$$\begin{aligned}
P(PosVeh \mid OPosVeh = \mathbf{e}) &= \frac{P(PosVeh,\ OPosVeh = \mathbf{e})}{P(PosVeh = \mathbf{e})} \\
&= \frac{P(PosVeh)\ P(OPosVeh = \mathbf{e} \mid PosVeh)}{\sum_{k=1}^{n} P(PosVeh = \mathbf{e},\ PosVeh_k)} \\
&= \frac{P(PosVeh)\ P(OPosVeh = \mathbf{e} \mid PosVeh)}{\sum_{k=1}^{n} P(OPosVeh = \mathbf{e} \mid PosVeh_k)\ \ P(PosVeh_k)}
\end{aligned} \tag{6.2}$$

With n the number of the node states.
This leads to an equation where all the elements have already being defined in the initialization of the prior probabilities.

Observation on the level of danger of the lanes (L_i) The nodes L_i are continuous observation nodes that describes the level of danger of the lanes with i being the lane number. It is based on an extended formulation of the TTC (ETTC) detailed in section 5.3.2.

In the MDL, the conditional probability distribution related to the ETTC under the condition of the status of the lanes $P(L_i|SL_i)$ is approximated by a normal distribution. This is justified by the fact that the ETTC is an estimation based on uncertain sensor observations and only a probability distribution is known [186, 212]. Thus, the likelihood function of the ETTC will be:

$$P(L_i \mid SL_i) = \int_{th_m}^{th_n} \mathcal{N}(\mu_{ETTC}, \sigma^2_{ETTC})(x)dx = \int_{th_m}^{th_n} \frac{1}{\sigma_{ETTC}\sqrt{2\pi}} \exp^{-\frac{1}{2}\frac{(x-\mu_{ETTC})^2}{\sigma^2_{ETTC}}}(x)dx \tag{6.3}$$

with $i \in \mathbb{N}$ (lane number), μ_{ETTC} is the mean and σ_{ETTC} is the standard deviation and x the corresponding value of the $ETTC$ for the corresponding lane L_i.

Most real-world problems involve the use of continuous quantities such as mass and temperature and by definition continuous variables have an infinity of possible values which makes specifying conditional probabilities for each value impossible. The common way to handle continuous variable is by using discretization which is dividing possible values into a fixed set of intervals. In this work, the collision level of risk is split into a five interval urgency rating: ETTC $\in [5,4[s$, ETTC $\in [4,3[s$, ETTC $\in [3,2[s$, ETTC $\in [2,1[s$, ETTC $\in [1,0[s$. The limits of integration th_n and th_m in equation (6.3) are based on these five-level discretization interval with th_n the upper bound of the interval and th_m the lower.

Based on a five-level discretization of the likelihood function $P(L_i|SL_i)$ and given uncertain evidence, equation (6.4) is obtained:

$$\begin{cases} P(L_i \mid SL_i = Dangerous) = \displaystyle\int_{th_m}^{th_n} \mathcal{N}(\mu_{ETTC} = \overline{ETTC_{dan}}, \sigma^2_{ETTC})(x)dx \\ P(L_i \mid SL_i = Occupied) = \displaystyle\int_{th_m}^{th_n} \mathcal{N}(\mu_{ETTC} = \overline{ETTC_{occ}}, \sigma^2_{ETTC})(x)dx \\ P(L_i \mid SL_i = Free) = \displaystyle\int_{th_m}^{th_n} \mathcal{N}(\mu_{ETTC} = \overline{ETTC_{free}}, \sigma^2_{ETTC})(x)dx \end{cases} \tag{6.4}$$

The value $\overline{ETTC_{state}}$ with $state$ = [Dangerous, Occupied, Free] represents the mean threshold for determining the level of dangerousness of the lane. The influence of the evidence on the probability distribution of the nodes is directly determined by the shape of the likelihood function and thus by the standard deviation. However, the standard deviation σ_{ETTC} in this work have been chosen to be the same for the three states and the approximation for each state will be studied in future work.

Status of lanes (SL_i) The nodes SL_i describe the status of occupancy of the lane. The possible states are *Dangerous* (vehicles present on the lane at a critical ETTC value to the ego-vehicle), *Occupied* (denoting the uncertainty and the risk outside of the critical zone) and *Free* (no vehicles present on the lane until a certain distance). This node is initialized to be uniformly distributed as we have no prior knowledge on the state of the node. The inference calculation for this node is then:

$$\begin{aligned} P(SL_i \mid L_i = \mathbf{e}) &= \frac{P(SL_i,\ L_i = \mathbf{e})}{P(L_i = \mathbf{e})} = \frac{P(SL_i)\ P(L_i = \mathbf{e} \mid SL_i)}{\sum_{k=1}^{n} P(L_i = \mathbf{e},\ (SL_i)_k)} \\ &= \frac{P(SL_i)\ P(L_i = \mathbf{e} \mid SL_i)}{\sum_{k=1}^{n} P(L_i = \mathbf{e} \mid (SL_i)_k)\ P((SL_i)_k)} \end{aligned} \tag{6.5}$$

6.3.2.2/ Decision-making in the MDL

The ultimate goal of the proposed decision making strategy is deriving the most suitable decisions given the available evidence. To identify the most suitable decision, we compute the Expected Utility (EU) for each decision state and the final decision is the alternative maximizing this EU. Appendix A details the basis of Multi-Level Decision Network (MLDN) and the EU equations.

Decision 1 (D_1) has four possible maneuvers:

- Lane Change Left (LCL) and Lane Change Right (LCR) for lane change maneuvers.
- Keep Lane with Adaptive Cruise Control (KL-ACC) for staying in the considered lane while keeping a safety distance with the ahead-vehicle.
- Maintain Velocity (MV) which is an alternative decision allowing to stay in the current lane while maintaining previous velocity configuration.

Utility nodes U_V defines the cost related to the decision. In the MDL (cf. Figure 6.3), U_S is the utility related to the safety of each of the maneuver alternatives activated and tailored by the ego-vehicle uncertain observations.
Its value is fixed deterministically to be zero or unity (unsafe or safe) depending on selected properties related to the traffic regulations or the common logic. For example, knowing that coming back to the origin lane is preferable after an overtaking, a preference is given to the decision to Lane Change Right if both lanes are safe.

Following the temporal order for this network $(SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane}) \prec D_1 \prec D_2 \prec D_3$, the EU for the first decision is D_1 given past observations $U_{Obs} = (SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane})$ is:

$$\begin{aligned}
&EU(D_1) = \sum_{U_{Obs}} P(U_{Obs}|\ D_1)\Big(U_S(SL_1, SL_2, PosVeh, D_1) + U_{Ch}(SM, D_2) + \\
&U_{Ev}(MaxCap,\ E_{Lane},\ D_3)\Big) = \sum_{U_{Obs}} P(SL_1)P(SL_2)P(Posveh)P(SM)P(MaxCap)P(E_{Lane}) \\
&\Big(U_S(SL_1, SL_2, PosVeh, D_1) + U_{Ch}(SM, D_2) + U_{Ev}(MaxCap,\ E_{Lane},\ D_3)\Big)
\end{aligned} \tag{6.6}$$

According to the Maximum Expected Utility (MEU) principle, the most suitable decisions are then:

$$\rho_1 = \max_{D_1} EU(D_1) \tag{6.7}$$

6.3.3/ Safety Verification Decision Level (SVDL)

The second decision is a part of the Safety Verification Decision Level (SVDL) where for each time step T_s, while the maneuver execution starts, a safety-checking regarding the action chosen in the MDL and a verification of the coherence of the maneuver with the predicted pre-planned trajectory is performed through an improved definition of anomaly detection criteria based-D-PIDP [2] (cf. Section 5.3.3.2), used to detect and compensate for possible failure or unexpected behaviors.

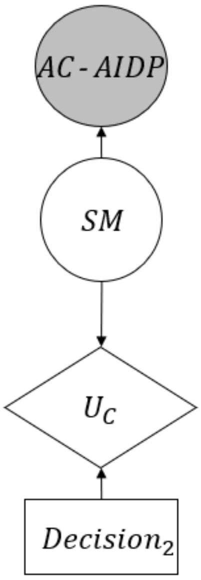


Figure 6.4: The Safety Verification Decision Level (SVDL) diagram

	Type of node	Description
AC-AIDP	Discrete Node	The observation on the anomaly detection criteria based on $t_{criticial}$
SM	Discrete Node	The status of the engaged maneuver based on the observation node

Table 6.3: The Safety Verification Decision Level (SVDL) variables

6.3.3.1/ Situation assessment variables in the SVDL

Anomaly detection criteria ($AC - AIDP$) Depending on the values of $t_{critical}$, defined in Section 5.3.3.2, input to the node AC-AIDP (cf. Figure 6.2), and for any vehicle pair that detects an anomaly, we will have two states:

- *Critical time positive* means that a value of critical time is found which means $d(t)$ goes beyond the limit safety boundary defined by $l(t)$ (cf. Figure 5.18).
- *No Anomaly Detected*.

The discrete AC-AIDP node constitutes a virtual binary node denoting the uncertain observation evidence input to node Status of Maneuver (SM).

Status of the maneuver (SM) This node describes the status of the engaged maneuver based on the observations that the node AC-AIDP provides. This node is initialized to be uniformly distributed as we have no prior knowledge on the state of the node.
The possible states are *Dangerous* (for the case where a critical time is found), *Safe* (the observation AC-AIDP does not endanger the situation).

6.3.3.2/ Decision-making in the SVDL

Decision 2 (D_2) has 2 states:

- *Abort Maneuver* (AM) that allow us to react to a dangerous change in the D-PIDP by advising to cancel the actual maneuver and to re-configure by computing an appropriate evasive maneuver (cf. subsection 6.4.3).
- *Maneuver is Safe* (MS) state that consolidates the previous decision made in node D_1 regarding to safety aspect.

Utility check U_{Ch} in the SVDL (cf. Figure 6.4), is the cost related to the safety verification during navigation based on the anomaly criteria.
Following the temporal order for this network $(SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane}) \prec D_1 \prec D_2 \prec D_3$, the EU for the second decision D_2 given given past observations $U_{Obs} = (SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane})$ and $D_1 = d_1$ is:

$$\begin{aligned}
&EU(D_2|\, d_1) = \sum_{U_{Obs}} P(U_{Obsv} \mid d_1, D_2)\Big(U_{Ch}(SM, D_2) + U_S(SL_1, SL_2, PosVeh, d_1) + \\
&U_{Ev}(MaxCap,\ E_{Lane},\ D_3)\Big) = \sum_{U_{Obsv}} P(SL_1)P(SL_2)P(PosVeh)P(SM)P(MaxCap)P(E_{Lane}) \\
&\Big(U_S(SL_1, SL_2, PosVeh, D_1) + U_{Ch}(SM, D_2) + U_{Ev}(MaxCap,\ E_{Lane},\ D_3)\Big)
\end{aligned} \tag{6.8}$$

According to the Maximum Expected Utility (MEU) principle, the most suitable decisions are then:

$$\rho_2 = \max_{D_2} EU(D_2|\, d_1) \tag{6.9}$$

The EADL will be detailed within the next section (cf. subsection 6.4.2), as an important work need to be performed before in order to present the elements of the evasive strategy.

6.4/ Evasive strategy

6.4.1/ Problem statement

In case of any anomaly is detected, if the initial suppositions to perform the maneuver are not anymore confirmed, decisions regarding possible evasive maneuver should be taken, to ensure the vehicles' safety (a literature review on this matter is proposed in Chapter 4, section 3.7.2). For this purpose, two main steps are defined:

1. The first step (cf. section 6.4.2) is preformed through the Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) where a third decision level called Evasive Action Decision Level (EADL) is proposed in order to choose one of the evasive action maneuver. It is computed relying on two observations [4].
    - The first one consists in computing the required deceleration a_{req}, following the definition of the critical time $t_{critical}$ (cf. Section 5.3.3.2 and Figure 5.18) and the

evolution of the distance descent (if an anomaly is detected). Computing the deceleration will allow us to asses if an emergency braking is possible given the actual situation configuration and given the vehicles' maximum capacity for braking a_{max}.

- The second observation consists in the endangered lane by finding the lane where $t_{critical}$ is positive (meaning one or more anomaly is detected in this lane).

In the first step, only the observations on deceleration and endangered lanes are used for the situation assessment in case of anomaly. If the emergency braking is not possible, other solutions are proposed by the EADL based on the observations.

2. The second step (cf. section 6.4.3) consists in adding a control module in the P-MCA architecture (cf. Figure 5.6, block 4) dedicated for the control part of the evasive maneuver. This module computes a low-level control sequence $\mathbf{u(t)} = (v(t), \gamma(t))^T$ (where $v(t)$ is the the linear velocity and $\gamma(t)$ is the steering angle of the vehicle) based on the evasive action maneuver advised by the Evasive Action Decision Level (EADL) of the SDN-MSV (step 1). This is performed while using a multi-criteria optimization based on an Evolutionary Algorithm (EA), the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (cf. Appendix B and section 6.4.3.2). The CMA-ES has been used to train the module to find the best possible strategy to drive the vehicle away from the dangerous configuration either by braking or by changing lane following the advise of $Decision_3$ (cf. Figure 6.7).

The reasoning have been performed in the evasive strategy for the case where the ego vehicle in in the right lane during a lane change maneuver. However, the reasoning is adaptable for other use-cases. The diagram given in Figure 6.5 illustrates the procedure for computing an evasive maneuver. This diagram is a focus of the *"Evasive Maneuver"* block in the overall flowchart of the whole decision-making strategy shown in Figure 6.1. The first step defined earlier is shown through blocks (2) and (3) of Figure 6.5 and will be detailed in section 6.4.2. The second step is shown through the blocks (4 to 7) and will be detailed in section 6.4.
A closed-loop action from the block (7) towards the initialization block (1) (cf. Flowchart given in Figure 6.1 for details about this block) of the algorithm is performed once the safety state is reached to restart the decision-making process.

6.4.2/ First step: The Evasive Action Decision Level (EADL)

6.4.2.1/ Criteria for evasive action selection

In the literature many criteria have been defined [32, 38, 124, 184] (cf. Section 3.7) to compute the remaining time span in which the driver can still avoid a collision by braking or by steering.
We proposed in [4] to compute the required deceleration a_{req}, based on the definition of the critical time $t_{critical}$ and the evolution of the distance descent (if an anomaly is detected), in order to choose one of the evasive action maneuver. Computing the deceleration will allow us to asses if an emergency braking is possible given the actual situation configuration and given the vehicles' maximum capacity for braking a_{max}. We first choose to check if the braking is possible before looking for other solutions.

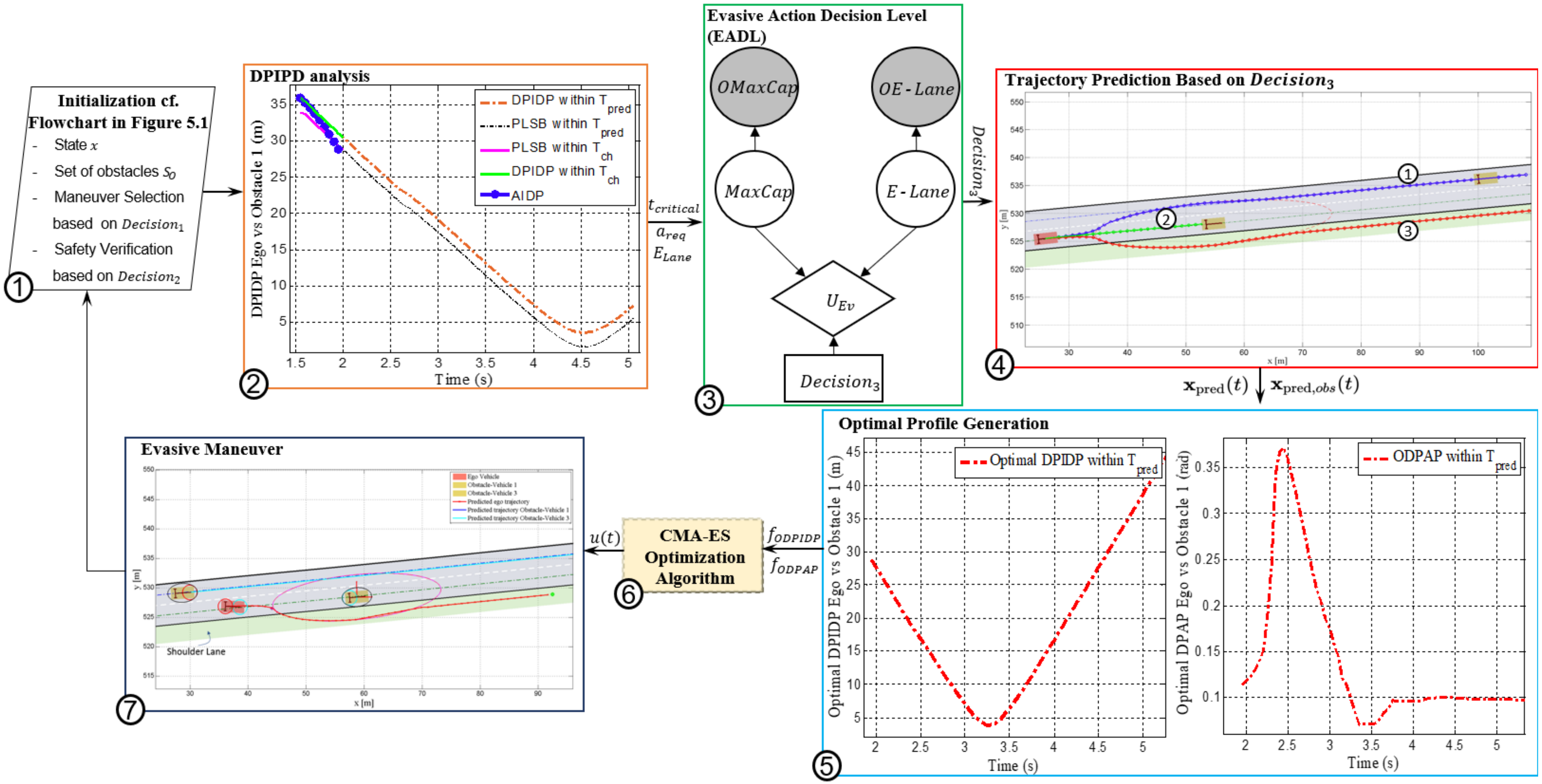


Figure 6.5: Overall procedure for computing an evasive maneuver

Assume that the ego-vehicle starts performing a lane change maneuver at an initial speed v_0 and that a change in the initial configuration happens (obstacle-vehicle 1 comes to standstill for example), meaning that $t_{critical}$ is positive (cf. Figure 5.18).
Given that we can know the distance drop d_{drop} (between the first variation of $d(t)$: $d_{|\xi|>\epsilon}$ and the intersection point t_{inter}, cf. Figure 6.6) caused by this change and that this distance drop happens during $t_{critical}$ and assuming a desired stopping inter-distance d_{stop} (between the ego-vehicle and the obstacle-vehicle 1) and a stopping velocity v_f at the end of the emergency braking (as the purpose is to avoid the collision), we are able to find the time required t_{req} for the ego-vehicle to reach d_{stop}, if d_{drop} remains constant, as:

$$t_{req} = \frac{d_{|\xi|>\epsilon} - d_{stop}}{d_{drop}} t_{critical} \tag{6.10}$$

Starting from the uniform acceleration equation : $v(t) = at + v_0$, we will have the required deceleration to reach d_{stop} defined by:

$$a_{req} = \frac{v_f - v_0}{t_{req}} \tag{6.11}$$

This criterion is used as a first step to resolve and propose an evasive solution by braking to our situation as will be seen in the following sections.

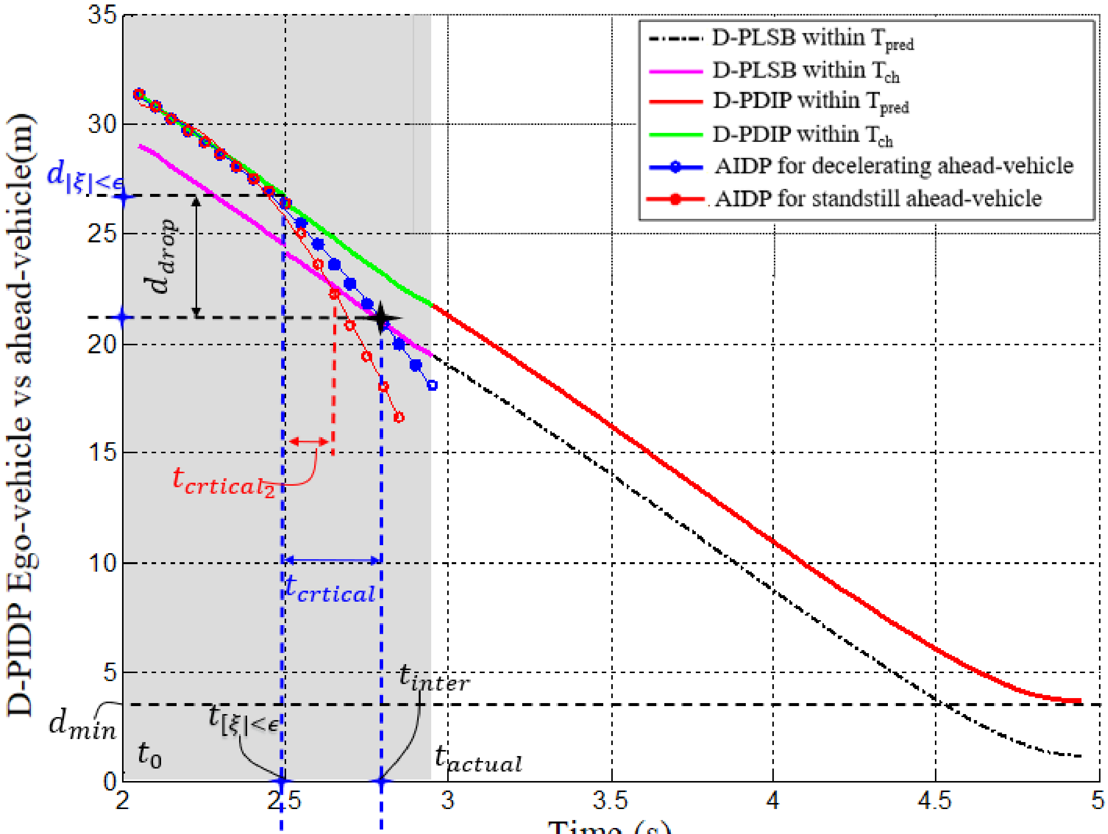


Figure 6.6: Definition of a_req based on the D-PIDP

6.4.2.2/ Situation assessment variables in the EADL

Using the above definitions, we are willing to we can define the following nodes as input to the EADL:

- *Observation on the ego-vehicles' Maximum Capacity ($OMaxCap$):* Depending on a_{req}, and on the definition of a virtual evidence (cf. subsection 6.3.2.1) two states are defined: $a_{req} \leq a_{max}$ and $a_{req} > a_{max}$

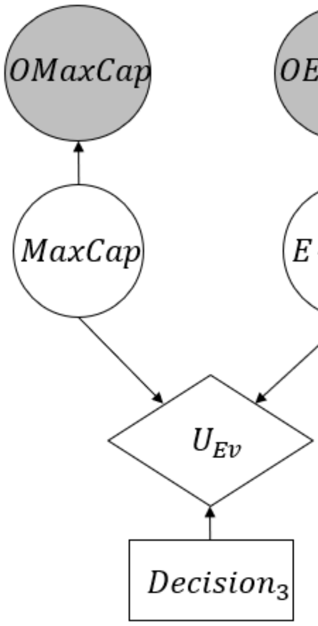


Figure 6.7: The Evasive Action Decision Level (EADL) diagram

	Type of node	Description
OMaxCap	Discrete Node	The uncertain observation on the maximum capacity of the ego-vehicle based on a_{req}
OE-Lane	Discrete Node	The uncertain observation on the endangered lane based on $t_{critical}$
MaxCap	Discrete Node	The maximum capacity of the ego-vehicle based the uncertain observation node
E-Lane	Discrete Node	The endangered lane based on the uncertain observation node

Table 6.4: The Evasive Action Decision Level (EADL) variables

- *Ego-vehicles' Maximum Capacity ($MaxCap$):* Depending on the observation node $OMaxCap$ provides, two states are defined: $a_{req} \leq a_{max}$ and $a_{req} > a_{max}$.

- *Observation on Endangered Lane ($OE - Lane$):* Depending on the values of $t_{critical}$ for each obstacle in each lane (if $t_{critical}$ is positive meaning one or more anomaly is detected in this lane) and for a road configuration of two lanes, this node has 3 states: Lane 1 is endangered, Lane 2 is endangered, Both Lanes are endangered.

- <u>*Endangered Lane ($E-Lane$):*</u> Depending on the observation node $OE-Lane$, 3 states are possible: Lane 1, Lane 2, Both Lanes on the lanes are endangered and emergency braking is not possible.

6.4.2.3/ Decision-making in the EADL

Decision 3 (D_3) in the other side, proposes 3 states for handling anomalies during lane change maneuver:

- *Continue Maneuver (CM):* in case for example only Lane 1 is endangered which means only the pair ego-vehicle/ahead-vehicle detects an anomaly ($t_{critical}$ is positive).
- *Emergency Braking (EB):* in case both lane are endangered which means $t_{critical}$ is positive for each pair of vehicles in each one of the lane and if the vehicles' maximum capacity for braking allows it $a_{req} \leq a_{max}$.
- *Emergency Stopping Lane (shoulder lane) (ESL):* in case where both lanes are endangered and emergency braking will not lead to safe situation (collision will happen).

The node Utility Evasive (U_{Ev}) is the cost related to the evasive action selection given its input $E-Lane$ and $MaxCap$.

Following the temporal order for this network $(SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane}) \prec D_1 \prec D_2 \prec D_3$, the EU for the third decision D_3 given past observations $U_{Obs} = (SL_1,\ SL_2,\ PosVeh,\ SM,\ MaxCap,\ E_{Lane})$ and $D_1 = d_1$ and $D_2 = d_2$, is defined as:

$$
\begin{aligned}
&EU(D_3|\, d_1, d_2) = \sum_{U_{Obs}} P(U_{Obsv} \mid d_1, d_2, D_3)\Big(U_{Ev}(MaxCap,\ E_{Lane},\ D_3) + U_{Ch}(SM, d_2) + \\
&U_S(SL_1, SL_2, PosVeh, d_1)\Big) = \sum_{U_{Obsv}} P(SL_1)P(SL_2)P(PosVeh)P(SM)P(MaxCap)P(E_{Lane}) \\
&\Big(U_S(SL_1, SL_2, PosVeh, d_1) + U_{Ch}(SM, d_2) + U_{Ev}(MaxCap, E_{Lane}, D_3)\Big)
\end{aligned}
\tag{6.12}
$$

According to the Maximum Expected Utility (MEU) principle, the most suitable decisions are then:

$$\rho_3 = \max_{D_3} EU(D_3|\, d_1, d_2) \tag{6.13}$$

Once the evasive discrete decision is taken by the EADL, let's now define the adopted strategy that allows to execute this decision while allowing smooth changes during the evasion and ensuring the safety of the system.

6.4.3/ Second step: Optimal low-level control based on CMA-ES for evasive maneuvers

The purpose of this section is to obtain an optimal control sequence $\mathbf{u(t)} = (v(t), \gamma(t))^T$ in order to follow the advised discrete evasive maneuver. This is performed while using a multi-criteria optimization based on the Covariance Matrix Adaptation Evolution Strategy

(CMA-ES) algorithm that has been used to train the module to find the best possible strategy to drive the vehicle away from the dangerous configuration either by braking or by changing lane following the advise of $Decision_3$ (cf. section 6.4.2). To this purpose we define two optimal safety profiles (cf. section 6.4.3.1): The Optimal Dynamic Predicted Inter-Distance Profile (OD-PIDP) and the Optimal Dynamic Predicted Angular profile (OD-PAP) calculated between each ego/obstacle combination. This evasive strategy is a proof of concept about the ability of the system to find an optimal distance/angular profiles to follow (instead of a trajectory) that allows us to find the best control sequence in order to perform the safe evasive action.

The diagram given in Figure 6.5 illustrates the procedure for computing an evasive maneuver. In what follows, the generated optimal profile (cf. block (4 and 5) in Figure 6.5, section 6.4.3.1) and the optimization strategy (cf. block (6 and 7) in Figure 6.5, section 6.4.3.2) will be detailed.

6.4.3.1/ Optimal safety feature selection: Predicted Profile Generation

The next step toward the evasive maneuver is to generate the OD-PIDP and OD-PAP. Before formalizing the optimization problem related to these profiles, let's define a set of criteria in order to generate them.

Criteria for optimal profile generation The optimal profile OD-PIDP and OD-PAP are generated based on some conditions:

- The Safety Verification Decision Level (SVDL) advise to abort the maneuver and the EADL computes the evasive decision.
- The generations of both profiles are based on the same concepts developed for the D-PIDP in Section 5.3.3. To do so, the adequate predicted trajectories have to be computed based on the evasive decision of the EADL. If for example $Decision_3$ advises to continue the maneuver, the trajectory predictions of the concerned vehicle pair according to this advise are calculated and both optimal profiles are generated and considered the ground truth regarding safety of maneuver (cf. Figure 6.8).
- These profiles are generated using the predicted trajectories that have the same velocity configuration at the time of the anomaly. This facilitates greatly the profile generation, however the optimization algorithm have to make sure to select a control sequence that respects the constraints defined in the optimization problem (regarding the rate of change of velocity and steering) while following at the very best the optimal profile (cf. Figure 6.5).
- The OD-PIDP profile satisfies a minimum distance requirement regarding the previous generated profile: $dist_{min}(t+1) \geq dist_{min}(t)$. This minimal distance during a lane change maneuver generally corresponds to the moment where the ego-vehicle is in the adjacent lane and the vehicles are side by side. This criteria is supposed to be satisfied by our dimensioning of the ellipse of influence (cf. section 5.2.2.2) during lane changes.
- Whether the vehicle performs a lane change during evasion to the left lane or to the emergency stopping lane, the algorithm verifies that the solution leads to the

center-line of the objective lane. This criteria is also supposed to be satisfied by our dimensioning of the ellipse of influence during lane changes.

An example of the resulting profiles is shown in Figure 6.8 following the conditions defined above.

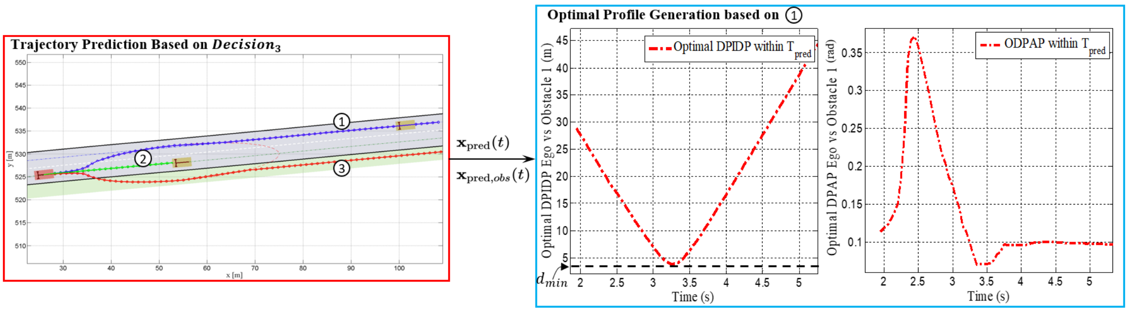


Figure 6.8: Optimal predicted profile generation

The second level decision is continuously updating during navigation and in case another anomaly is detected and we are not able to generate a profile that respects the above condition in this case we may consider collision mitigation [1]. This topic will be considered for future work.

Formalization of the objective function related to to OD-PIDP The motion of the ego-vehicle is described by a tricycle model as shown in equation (5.7). In what follows $\mathbf{X} = \{x, y, \theta\}$ is the state vector with $(\mathbf{x}, \mathbf{y})$ the vehicle's position and θ its orientation, v and γ are output of the control law (cf. subsection 5.1.5) representing the velocity and the steering angle respectively, l_b is the wheel-base of the vehicle.

Based on Euler's Method to solve first order differential equation with a given initial value, we can write:

$$\begin{cases} x(t+h) = x(t) + h\ v(t)\ cos(\theta(t)) \\ y(t+h) = y(t) + h\ v(t)\ sin(\theta(t)) \\ \theta(t+h) = \theta(t) + h\ v(t)\ tan(\gamma(t))/l_b \end{cases} \tag{6.14}$$

with $t \in [t_0, T_{pred}]$ and h the time step size.

The motion of the surrounding obstacle-vehicles is assumed to be rectilinear uniformly accelerated and is described by the following equations:

$$\begin{cases} x_{obs}(t+h) = x_{obs}(t) + \frac{1}{2} a_{x_{obs}}(t) h^2 + v_{x_{obs}} h \\ y_{obs}(t+h) = y_{obs}(t) + \frac{1}{2} a_{y_{obs}}(t) h^2 + v_{y_{obs}} h \end{cases} \tag{6.15}$$

With (x_{obs}, y_{obs}) the obstacle-vehicle's position, $(v_{x_{obs}},\ v_{y_{obs}})$ the speed components and $(a_{x_{obs}},\ a_{y_{obs}})$ the acceleration components.

The formalization of an inter-distance prediction profile can be defined as the function $p(t+h)$ over the interval $t \in [t_0, T_{pred}]$:

[1] Actions in order to reduce at maximum the injures of the passengers, and/or the others outside the ego-vehicle [98]

$$
\begin{aligned}
p(t+h) &= \Big(\big(x(t+h) - x_{obs}(t+h)\big)^2 + \big(y(t+h) - y_{obs}(t+h)\big)^2\Big)^{1/2} \\
&= \Big(\big(x(t) + hv(t)cos(\theta(t) + hv(t)\frac{tan(\gamma(t))}{l_b}) - x_{obs}(t) - h^2\frac{1}{2}a_{x_{obs}}(t) \\
&- hv_{x_{obs}}(t)\big)^2 + \big(y(t) + \ hv(t)cos(\theta(t) + \ hv(t)\frac{tan(\gamma(t))}{l_b}) - y_{obs}(t) \\
&- h^2\frac{1}{2}a_{y_{obs}}(t) - hv_{y_{obs}}(t)\big)^2\Big)^{1/2}
\end{aligned}
\tag{6.16}
$$

By analyzing the following formalization in equation (6.16), we can see that it highlights the needed sequence $\mathbf{u(t)}$. This formulation allows to have convenient way to define for each ego-vehicle/obstacle combination, an error objective function of the inter-distance between the reference O-DPIPD and the prediction $p(t+h)$ when applying the control sequence $\mathbf{u(t)}$ at a given time, and is defined as follows :

$$
f_{ODPIDP}(t) = \mid p(t+h) - ODPIDP(t+h) \mid \quad \text{for } t \in [t_0,\ T_{pred}] \tag{6.17}
$$

Formalization of the objective function related to OD-PAP The need to formalize an angular profile arised after the first test while using only the OD-PIDP. This test showed us that the vehicle is able to find a solution $u(t)$ that follows the defined OD-PIDP while respecting the defined set-points but goes outside the road. In conclusion, as obvious as it may seem the vehicle needs to respect not only the predicted inter-distance profile but also a certain angular profile to be able to stay within the road range. For this reason, the D-PAP will be generated accordingly following the criteria presented in section 6.4.3.1

The formalization of an angular prediction profile, defined as function $\theta(t+h)$ (cf. equation (6.14)) over the interval $t \in [t_0, t_{pred}]$, that highlight the concerned control sequence $u(t)$ is then:

$$
\theta(t+h) = \theta(t) + \frac{T_s v tan(\gamma)}{l_b} - \theta_{Obs} \tag{6.18}
$$

We assumed in this work that motion of the surrounding obstacle-vehicles is rectilinear uniformly accelerated, which makes the heading of the concerned obstacle-vehicle θ_{Obs} directly related to the curvature of the road.
Similarly to the OD-PIPD, the strategy is to minimize the absolute value of the error between the reference O-DPAP and the prediction $\theta(t+h)$ when applying the control sequence $\mathbf{u(t)}$ at a given time. The used error objective function will be defined as:

$$
f_{ODPAP}(t) = \mid \theta(t+h) - ODPAP(t+h) \mid \quad \text{for } t \in [t_0,\ T_{pred}] \tag{6.19}
$$

6.4.3.2/ MULTI-OBJECTIVE AND CONSTRAINTS FUNCTION

The optimal sequence is defined as the one that minimizes a global function that combines both the objective functions related to OD-PIDP and OD-PAP. This global multi-objective function punishes high accelerations and high curvature rates with the weights

$w_d \in \mathbb{R}_+$ and $w_a \in \mathbb{R}_+$, and is defined as the following :

$$J[u(t)] = \int_{t_0}^{t_0+T_h} F[u(t)]dt, \tag{6.20}$$

with

$$F[u(t)] = \sum_{i=1}^{n_{obstacles}} \Big(w_{d_i} \, f_{ODPIDP_i} + \, w_{a_i} \, f_{ODPAP_i} \Big). \tag{6.21}$$

The time t_0 is the current time, T_h is the time horizon in the interval $[t_0, \ T_{ch}]$ and i is the obstacle's Id number. Precise analysis of the appropriate balance between each sub-criteria w_d and w_a will be the subject of future work.

The optimal sequence must minimize the function described by equation (6.20) and at the same time obey to a set of defined constraints. This constraints results from the limits of the vehicle kinematics and dynamics.
The steering input angle is limited by the steering geometry of the vehicle concerning the steering lock angle and the steering rate of change as we aim at minimizing J and punishing high curvature rates to achieve smooth trajectories, so:

$$\begin{gathered} -\gamma_{max} \leq \gamma(t) \leq \gamma_{max} \\ |\dot{\gamma}(t)| \leq \dot{\gamma}_{max} \end{gathered} \tag{6.22}$$

The vehicle is also bounded concerning its applied velocity and in the rate of change of the velocity to favor comfortable trajectories by punishing high accelerations and jerk, so:

$$\begin{gathered} v_{\min} \leq v(t) \leq v_{\max} \\ a_{\min} \leq a(t) \leq a_{\max} \end{gathered} \tag{6.23}$$

The proposed strategy will allow the overall control architecture to increase its degrees of freedom concerning the maneuverability of the vehicle, allow smooth changes during the evasion, and ensuring the safety of the system and respecting as much as possible the passengers comfort.

6.4.3.3/ Solving the optimization problem based on CMA-ES

This optimization problem is solved using an evolutionary algorithm called the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). A detailed description of the main properties of the used Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm are given in Appendix B. An open source code [114]and a tutorial provided by the INRIA [92] have been used to implement the CMA-ES in this work.

Very few modification have been introduced to the original algorithm as the strength of this algorithm is that the CMA-ES does not require a tedious parameter tuning and the choice of internal parameters of the strategy is not left to the user except for population size. Some test have been performed in order to reduce the computation time by reducing the population size and are shown in the following simulation in Table 6.5.
The algorithm takes as input the defined multi-objective function, the initial velocity/steering configuration, the weights and constraints thresholds. The time horizon T_h in the other size have been fixed to be equal to the sampling time T_s. This was sufficient to converge to the solution as the optimization was preceded with an optimal construction

of the profiles that was accurate enough. The stop condition of the optimization was fixed when condition6.4.3.1 of section 6.4.3.1 was satisfied.
This evasive strategy is a proof of concept about the use of optimal profiles to follow, that needs to be refined and optimized particularly regarding the computation time and the prediction horizon.

6.5/ Simulation results of the proposed decision-making strategy

To evaluate the presented approach, a MATLAB car simulator has been developed. The simulator was used to generate all the needed environment to test the developed algorithms.

For the different simulations shown below, it is considered what follows:

- The perceived scene is constituted of four vehicles in a two-lane highway: two vehicles on the right lane (named respectively ego-vehicle and obstacle-vehicle 1 O_1) and two vehicles on the left lane (named respectively obstacle-vehicle 2 O_2 and obstacle-vehicle 3 O_3). The setup of the simulation environment is shown in Figure 6.9.

- The initial velocities of the vehicles are given by: $V_{ego_{max}} = 30m/s$, $V_{O1} = 12m/s$, $V_{O2} = 25m/s$ $V_{O3} = 20m/s$.

We will start to show the abilities of the proposed P-MCA in a nominal traffic condition in the following simulations.

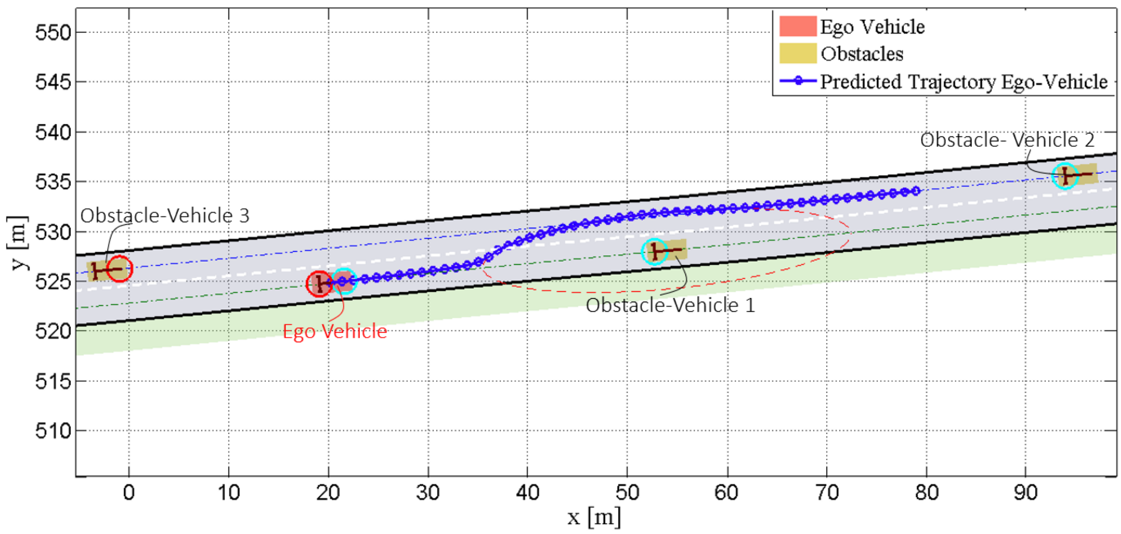
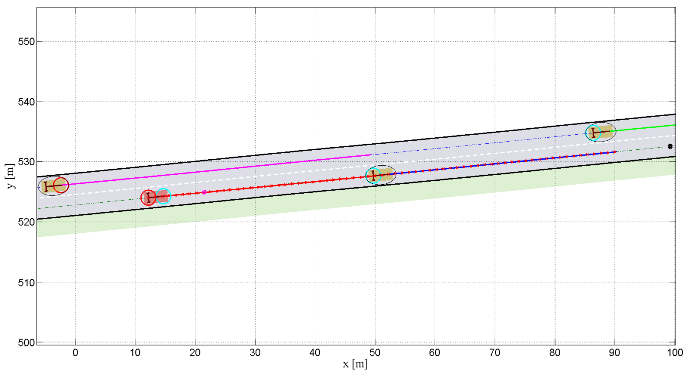


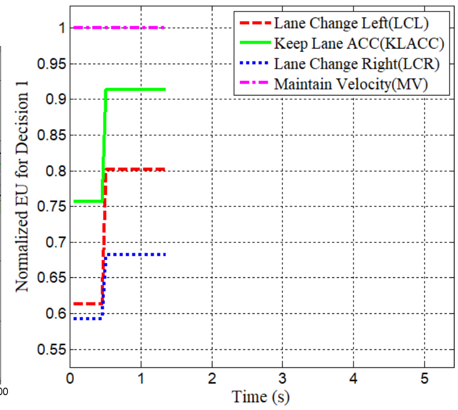
Figure 6.9: Setup of the simulation environment

6.5.1/ Demonstrative example in nominal traffic condition

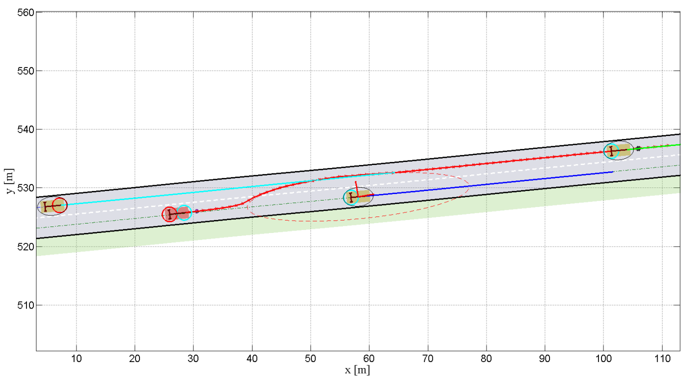
In this demonstrative simulation, as the ego-vehicle approaches the obstacle-vehicle, it performs an overtaking maneuver with Lane Change Left (LCL) to the left. Figure 6.10 shows the EU of each decision alternatives for both decisions: at the beginning of the scenario the ego-vehicle approaches the obstacle-vehicle in front which leads the EU of the Maintain Velocity (MV) state to decrease while the EU of state LCL increases. This is justified by the fact that the distances are shrinking and SL_1 is approaching the



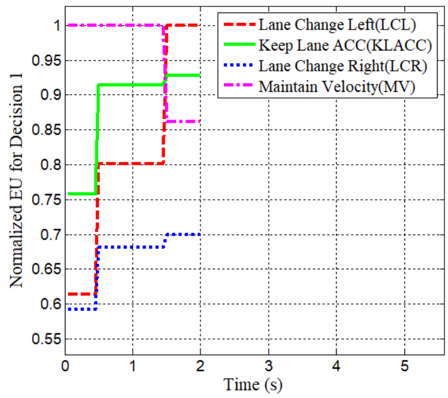
(a) The ego-vehicle keeps its lane



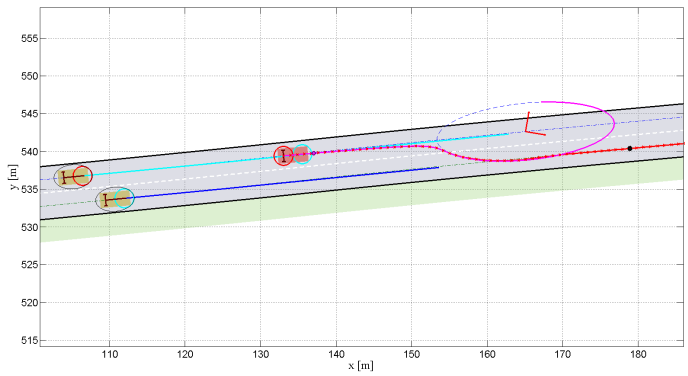
(b) $Decision_1$ is MV



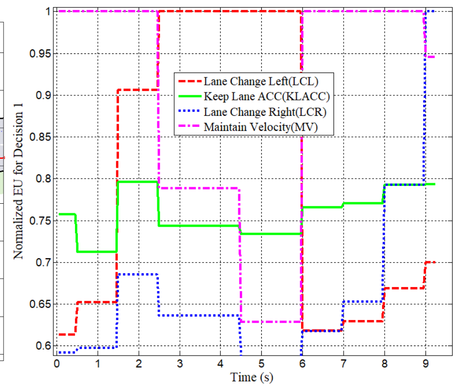
(c) Overtaking maneuver of the vehicle in front



(d) $Decision_1$ is LCL



(e) Lane change right



(f) $Decision_1$ is LCR

Figure 6.10: Snapshots of the car simulator during normal condition navigation and the corresponding Maneuver Decision Making

state *Occupied*. Since $Decision_1$ is in the state *LCL* the decision in node $Decision_2$ is simultaneously computed while taking into account the D-PIDP. In this case, the most suitable decision is *Maneuver is Safe (MS)* and remains constant during the whole lane change maneuver.

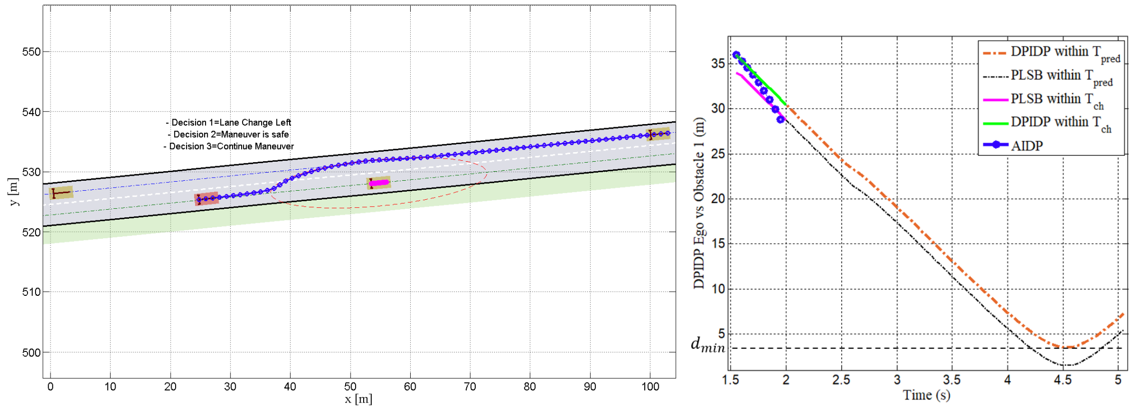
After the lane change part is achieved, the vehicle is now on the left lane, a decision to *Maintain the velocity* is activated to outrun the obstacle-vehicle 1 that is overtaken. Simultaneously, while maintaining the actual velocity, the ETTC value calculated between the ego-vehicle and the obstacle-vehicle 2 initially present in the left lane is dropping. Sometime later, the ego-vehicle is at a convenient TTC value from the obstacle-vehicle 1 located in the right lane and a *Lane Change Right (LCR)* is performed as it is the alternative with the highest EU.
In order to perform a Lane Change to the right we use the same formalism defined for an overtaking maneuver. Given that the purpose is to come back to the original lane, a virtual point is used to be the center of the ellipse of influence in order to generate the limit-cycle trajectory as shown in Figure 6.10e.

In the next sections simulations scenarios will be shown in emergency situations.

6.5.2/ Demonstrative examples in emergency situations

6.5.2.1/ Scenario 1 - Lane 1 is endangered

In what follows, we have selected a dangerous scenario where the obstacle-vehicle 1 in front suddenly brake, while the ego-vehicle is trying to perform a lane change maneuver (cf. Figure 6.11a). Both $Decision_1$ and $Decision_2$ are recomputed simultaneously at each time step at this stage following the defined flowchart (defined in Figure 6.1, Section 6.2). In this case, we can see in Figure 6.11b that the Actual Inter-Distance Profile (AIDP) crosses the Dynamic Predicted Lower Safety Boundary (D-PLSB) generating consequently the Safety Verification Decision Level (SVDL) to advise aborting the maneuver given that $t_{critical}$ is positive.
In this case, given that the left lane is free and given the ability of the system to reconfigure and adapt to the change (thanks to the properties of the Elliptic Limit-Cycle (ELC) (cf. subsection 5.2.2.2) and to the Dynamic Predicted Inter-Distance Profile (D-PIDP)) the evasive action maneuver is to continue the lane change maneuver with the appropriate settings (adaptation of the ellipse dimensions to the actual velocity/position configuration). The sequencing of decisions at this stage is shown in Figure 6.12. Following the reasoning defined in previous chapters and following procedure for computing the evasive maneuver (cf. Section 6.4 and Figure 6.5), the trajectory predictions are performed according to the evasive decision and the optimal profiles are generate. The CMA-ES then computes the control sequence that allows to follow as accurately as possible the defined profiles as shown in Figure 6.13.
Meanwhile, the SVDL supervise the procedure by continuously calculating the criteria given the optimal profile making sure that the profile is well followed. The overall steering and velocity profiles are shown in figure 6.14. After the defined horizon passes and the vehicle completed its maneuver (which corresponds in this case is the lane change maneuver, cf. Section 5.2.2.3 for definition of a completed lane change maneuver), the system recalculates $Decision_1$ in order to pursue the navigation.



(a) Configuration of the vehicles

(b) D-PIDP for the vehicle pair: Ego-vehicle vs Obstacle-Vehicle 1

Figure 6.11: Configuration at the moment of the anomaly for Scenario 1

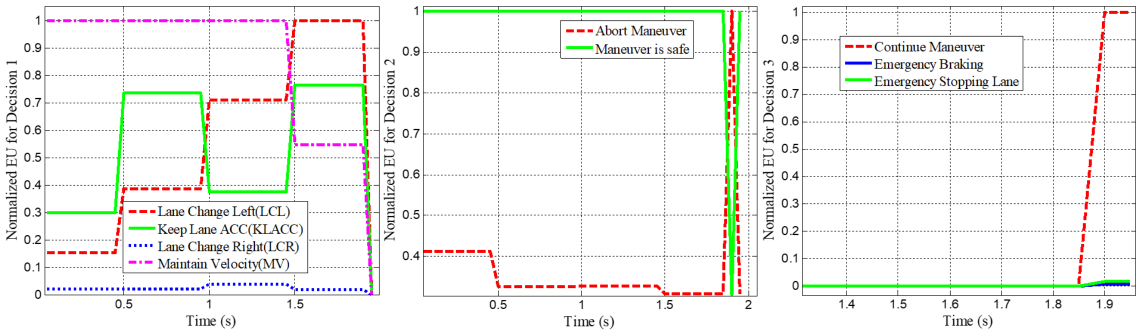
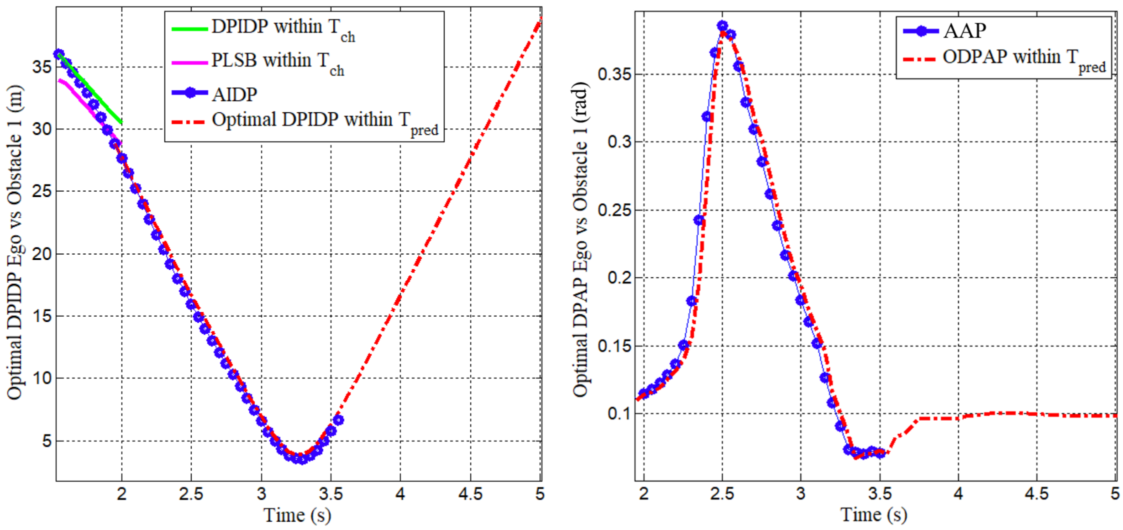


Figure 6.12: Sequencing of Decisions in emergency situation for Scenario 1



(a) Optimal D-PIDP

(b) Optimal D-PAP with AAP the Actual Angular Profile

Figure 6.13: Generated Optimal Safety Profiles in emergency situation for Scenario 1

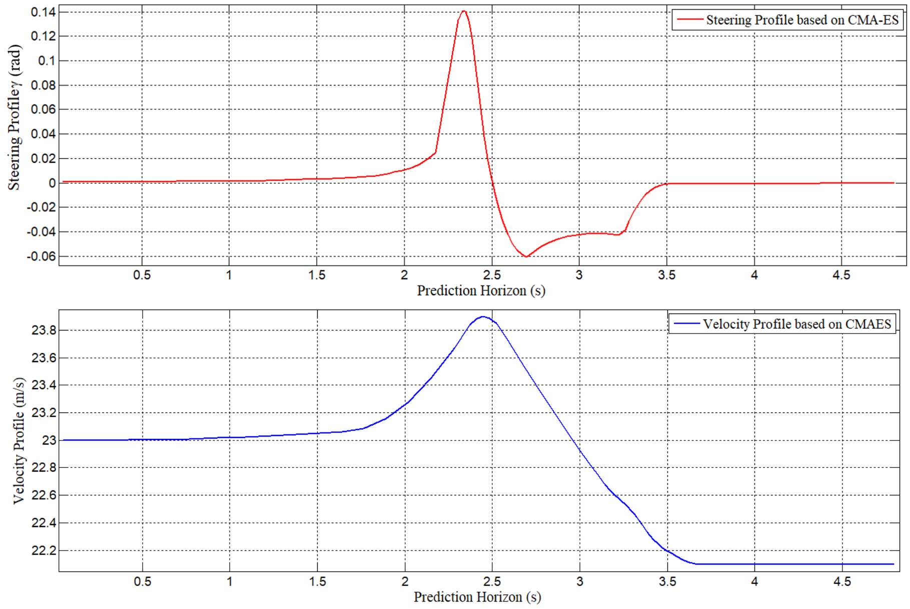


Figure 6.14: Steering and velocity profiles during the navigation for Scenario 1

6.5.2.2/ Scenario 2 - Lane 1 and Lane 2 are endangered

In order to go one step further we simulated a sudden acceleration of the obstacle-vehicle 3 coming from behind in the left lane. At the beginning of the simulation this obstacle is far and slow enough to allow the lane change maneuver to start but suddenly accelerates. Consequently, two of the three D-PIDP profiles, corresponding to obstacle-vehicle 1 and obstacle-vehicle 3 alert us through the anomaly criteria (cf. section 5.3.3.2) that the current situation is dangerous and the lane change maneuver is impossible.
In this case the appropriate decision is to abort the maneuver (cf. Figure 6.15) and two different evasive maneuver are possible: Emergency braking or Emergency Stopping Lane. The Emergency Braking is possible if and only if $a_{req} \leq a_{max}$. In this second studied scenario (Scenario 2), $a_{req} > a_{max} = -10m/s^2$, which leads the system to choose the Emergency Stopping Lane as discrete evasive action (cf. Section 6.4.2). The sequencing of decisions at this stage is shown in Figure 6.15.
Following the reasoning defined in previous chapters and following procedure for computing the evasive maneuver (cf. Figure 6.5), the predictions (shown in Figure 6.16) are performed according to the evasive decision and the optimal profiles are generated.

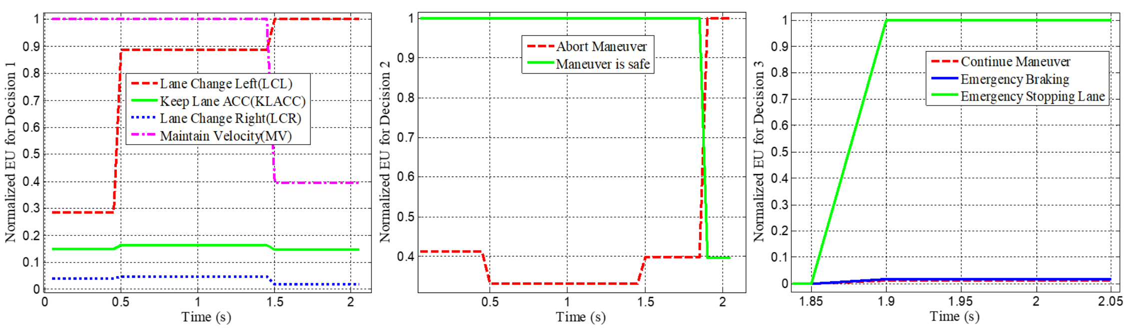


Figure 6.15: Sequencing of Decisions in emergency situation for Scenario 2

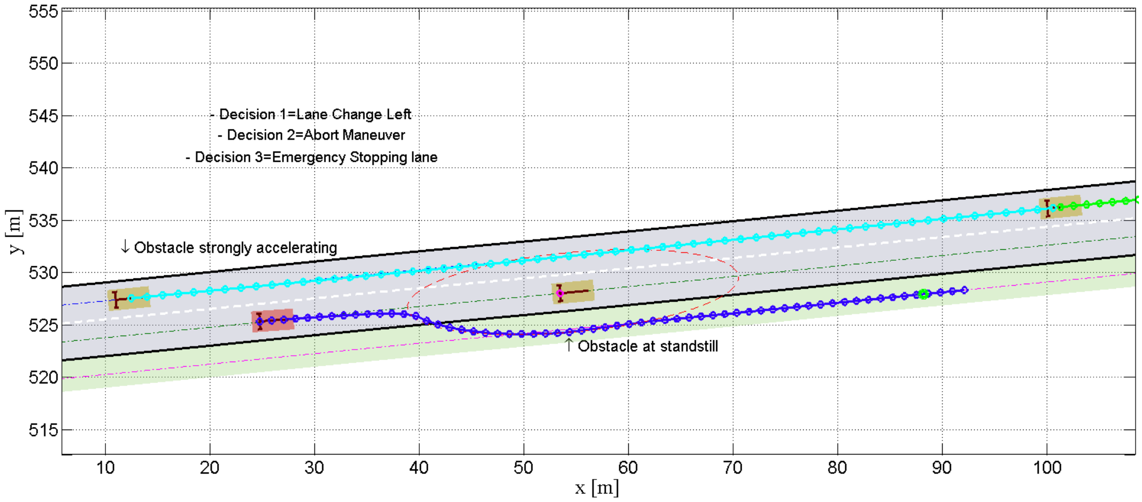
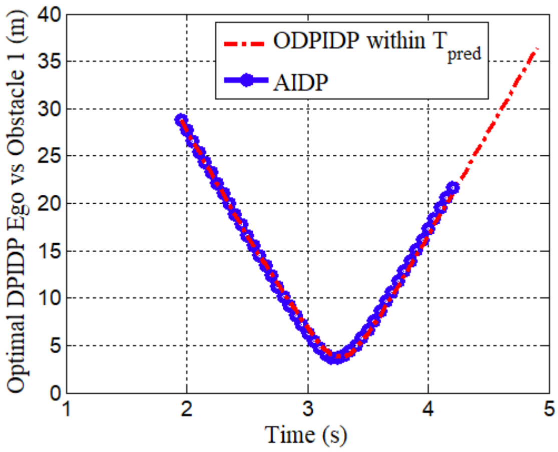
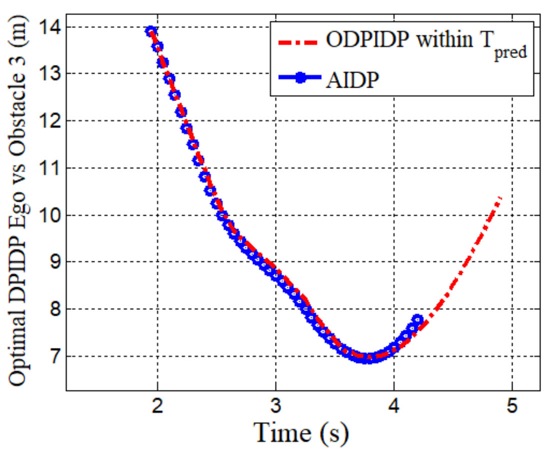


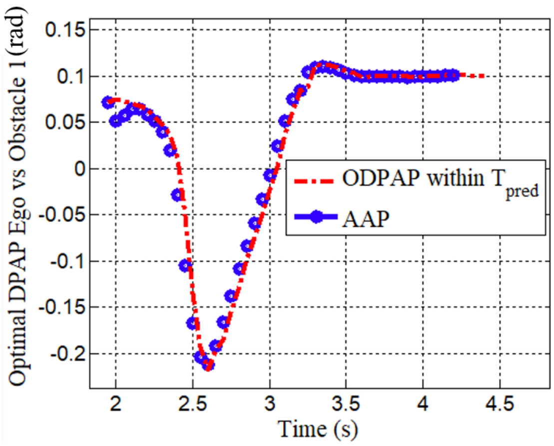
Figure 6.16: Evasive trajectory involving swerving to the shoulder lane for Scenario 2



(a) Optimal D-PIDP: Ego vs. Obstacle-Vehicle 1



(b) Optimal D-PIDP: Ego vs. Obstacle-Vehicle 3



(c) Optimal D-PAP

Figure 6.17: Generated Optimal Safety Profiles in emergency situation for Scenario 2

The CMA-ES then computes the control sequence that allows to follow as accurately as

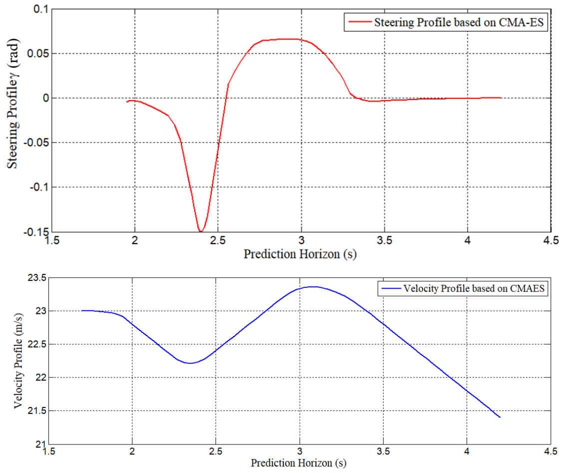


Figure 6.18: Steering and velocity profiles during the evasive maneuver for Scenario 2

possible the defined profiles as shown in Figure 6.17.
The overall resulting steering and velocity profiles are shown in figure 6.18. We can notice that at the beginning the evasive trajectory do not compromise comfort since the vehicle have to swerve to the shoulder lane while a lane change maneuver to the Left is starting.

In the other case, if the deceleration of the obstacle-vehicle 1 is smoother, $a_{req} \leq a_{max} = -10m/s^2$. This induce the third decision to be emergency braking and in this case applying a_{req} on the ego-vehicle is sufficient to guarantee safety since the longitudinal constraints required here is already satisfied by the procedure to deduce a_{req} (cf. Section 6.4.2.1 and Figure 6.19).

Table 6.5 summarizes the different parameters taken for each of the above scenario. The angular profile have a bigger weight as we argue for the experiment of these simulations that a bad turn of the steering wheel can cause the vehicle to deviate quickly out of its path and correcting it can become very difficult. The proposed methodology can also be extended for cases where obstacle-vehicles change lane abruptly in configurations where a collision might occurs. This precise point will be discussed in the perspectives

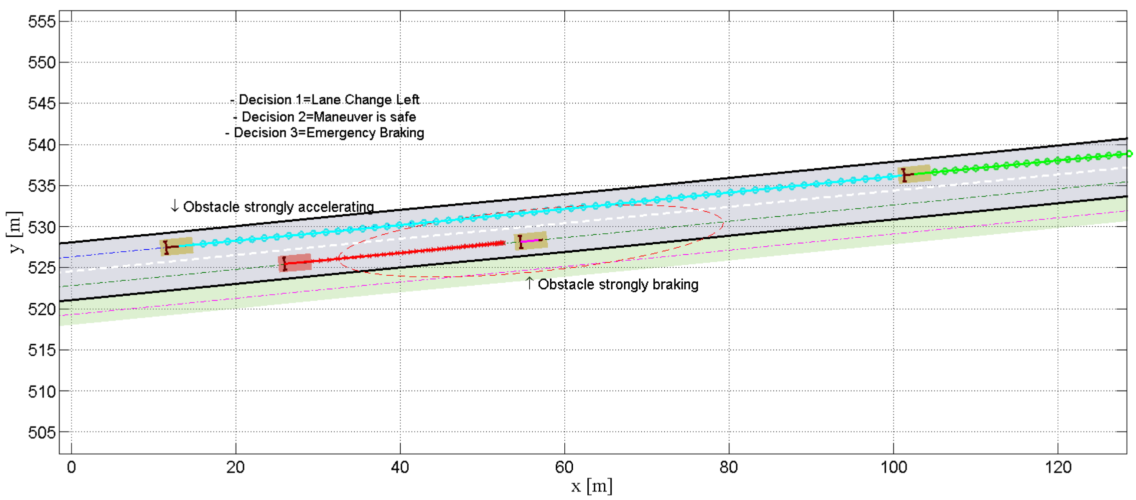


Figure 6.19: Evasive trajectory involving braking

Scenario	Avg. computation time	Weights	Nbr. of Generation
Lane 1 is endangered	$t = 0.05$s	$w_{a_1} = 2,\ w_{d_1} = 1$	5
Lane 1 and 2 are endangered	$t = 0.07$s	$w_{a_1} = 3,\ w_{d_{1,3}} = (1, 2)$	5

Table 6.5: The CMA-ES parameters

of this Ph.D. as we assumed in this work a rectilinear uniformly accelerated motion from the obstacle-vehicles. In any other cases, where the optimization is not able to reach a solution (the lateral and longitudinal constraints would be violated for example) and a collision cannot be avoided, collision mitigation should be applied. These kind of scenarios will be subject of future works. Moreover, the evasive strategy is a proof of concept about the use of optimal profiles to follow, that needs to be refined and optimized particularly regarding the computation time and the prediction horizon.
For the different scenarios stated above, videos has been realized for each scenario and are available through this link: https://shorturl.at/isvI4.

6.6/ Conclusion

In this chapter, a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) is proposed as a decision making framework. It corresponds to the main module of the proposed Probabilistic Multi-Controller Architecture (P-MCA).
This module is designed to manage several road-way maneuvers under uncertainties (which are due mainly to perceptive and/or other vehicles' intention/actions lack of precision). It utilizes multiple complementary threat measures (detailed in chapter 5 Section 5.3) in order to propose discrete actions that allows to: derive appropriate maneuver in a given traffic situation, provide a safety retrospection and verification over the current maneuver risk and and take evasive action according to the possible extreme situations faced to the autonomous ego-vehicle. The first discrete action concerns the choice of action regarding the most suitable maneuver and it is based on the current situation assessment, using the Extended Time-To-Collision (ETTC). The second discrete action is to decide whether or not the ongoing maneuver is to be aborted. This decision is based on a safety verification regarding the action chosen in the first decision and is used to detect and compensate for unexpected behaviors such as other objects and road users entering the planned course of the vehicle. It is based on a defined anomaly criterion through the definition of the Dynamic Predicted Inter-Distance Profile (D-PIDP). The third discrete action concerns the possibility for the vehicle to re-plan its path in case the maneuver is aborted by determining an alternate route, i.e., the emergency trajectory. This decision is based on the vehicles maximum capacities and on the endangered lane. Furthermore, based on the third decision output of the SDN-MSV, it is proposed to compute the corresponding low-level control that allows the AV to pursue the advised evasive maneuver to avert an accident and guarantee even more safety of the AV. This is accomplished by defining optimal safety profiles that should be matched to stay safe. This is performed while using an optimization algorithm based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Finally, simulations were shown in this chapter to validate the overall architecture in nominal and emergency situations (cf. section 6.5).

In the next chapter, the performed experiments and the used tools for the implementation of the P-MCA on experimental vehicles will be shown.

7

EXPERIMENTAL EVALUATION OF THE P-MCA AND DEMONSTRATIONS

Contents

The Probabilistic Multi-Controller Architecture (P-MCA) is validated both in simulated traffic in various situations (presented in sections 5.2.3 and 6.5) and on an experimental platform PAVIN (**P**lateforme **A**uvergnate pour les **V**éhicules **IN**telligents).

Simulation has the advantage of being able to generate a large number of scenarios of vehicles involved in dangerous configurations. However, a simulated environment is not fully representative of real-life situations notably concerning the perception capabilities that are assumed to be perfect. On the other hand, the difficulty in testing decision-making frameworks in real situations is that dangerous scenarios are difficult to reproduce with real vehicles because it can be dangerous and costly if an accident happens. In addition, collecting a database in this domain with complex configuration would take a tremendous amount of time since these are rare events. Even in the case where we want to test these scenarios, a good preparation of the environment is needed.

For this reasons, the strategy adopted in this work was to conduct from one side pure simulations to validate the overall architecture in dangerous and not dangerous situations (presented in sections 5.2.3 and 6.5) .
And on the other side, we validate the online functioning of the algorithm as well as its robustness to imperfect input data in real situations. This chapter describes the performed experiments and the used tools for the implementation of the Probabilistic Multi-Controller Architecture (P-MCA) for autonomous navigation.

7.1/ USED PLATFORMS FOR FIELD TRIALS

The electrical urban vehicle **IP**car (**I**nstitut **P**ascal cars) from APOJEE company [116] used in our experiments is a platform dedicated to the development of autonomous vehicles. They have been used to implement several proposed control architectures for autonomous navigation of mono- or multi-vehicle navigation [9, 219]. The IPcar carries different embedded proprioceptive and exteroceptive sensors. It has cameras, RTK-GPS, odometers, IMUs, lidars, a Wi-Fi communication system and an embedded computer (cf. Table 7.1 for some IPcar's main specification). The IPcar can be controlled using the onboard computer (through CAN protocol) or while using the wired control panel attached to the vehicle.

The test platform named Plate-forme d'Auvergne pour Véhicules INtelligents (PAVIN) is located at Campus Cézeaux of Clermont Auvergne University in Clermont-Ferrand (cf. Figure 7.2). PAVIN is an artificial environment composed by two areas (urban and rural areas) which have a total ground surface of 5.000 m^2 which serves as a testbed for mobile robotic applications. The urban area has a trajectory of 317 m containing scaled street with several traffic junctions and roundabouts with traffic sign boards and lights wherever necessary. Moreover, building facades on both sides, vegetation and street furniture are set to bring to a whole scene. The rural area has a trajectory of 264 m with unpaved roads, grass and mud on the roadsides. In addition, the whole area is covered by a wireless network and a DGPS base station [116]. Although PAVIN is a small scale environment, it stands as an efficient platform for evaluating algorithms related to autonomous driving such as navigation, road detection, traffic signal detection, etc. A 2D and a textured 3D model of PAVIN environment geo-referenced with high-precision GPS data are available in [116].

IPcar	Elements	Description
	Chassis	$(l, w, h) = (1.96, 1.30, 2.11)\ m$
	Weight	400 kg (without batteries)
	Motor	Triphase 3x26 V, 4KW
	Break	Integrated to the motor
	Max. speed	20 km/h ($\approx\ 5.5 m/s$)
	Batteries	8 batteries 12 V, 80 Ah
	Autonomy	3 hours at full charge
	Computer	Intel Core i7, CPU: 1.8 GHz RAM:16GB, OS: Ubuntu Mate 16.04

Table 7.1: IPcar's specifications

Several sensors are mounted in the VIPALAB to obtain information about the other AV or environment (cf. Figure 7.1). The main sensors used for experiments are described in Table 7.2.

7.2/ USED SOFTWARE TOOLS

Several interesting simulators exist in the field of autonomous driving [179]. Table 7.3 summarizes some of them.

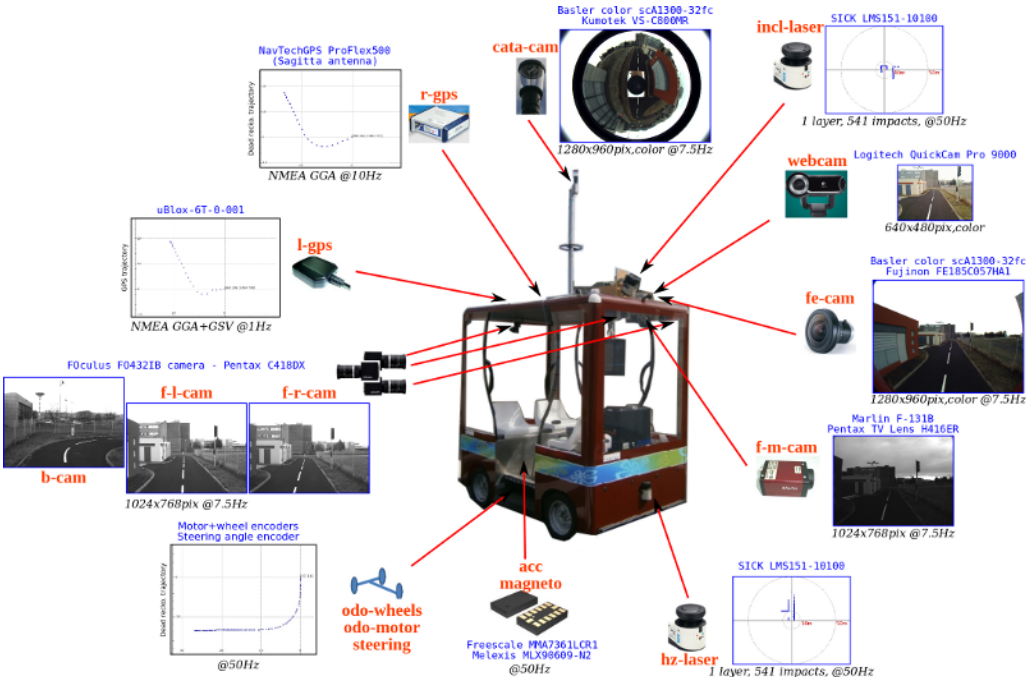


Figure 7.1: The IPcar with all sensors with their mounting locations and characteristics. [Image Credit:[116]]

Elements	Description
RTK-GPS	NacTechGPS, accuracy: $2\mathrm{cm}$; framerate: $10\mathrm{Hz}$
IMU	Xsens MTi, accuracy: $0.2°/\mathrm{s}$; framerate: $2\mathrm{KHz}$
Range sensor	SICK LMS, range [0,50] m and angle $[-45°, 225°]$, resolution: $0.5°$ framerate: $50\mathrm{Hz}$
Proprioceptive sensor	Wheel odometry, accuracy: $2\mathrm{cm}$; framerate: $50\mathrm{Hz}$
	Steering angle, resolution: $0.02°$; framerate: $50\mathrm{Hz}$
	Motor odometry, resolution: $0.1\mathrm{m/s}$; framerate: $50\mathrm{Hz}$

Table 7.2: IPcar's sensors

Figure 7.2: PAVIN experimental platform (Clermont-Ferrand, France)

Simulator	Description
CARLA	Urban simulator, Camera & LIDAR streams, with depth & semantic segmentation, Location information
CarMaker	testing for the fields of autonomous driving, ADAS, powertrain and vehicle dynamics, portfolio of sensor models available to facilitate the validation of automated driving functions, realistic test scenarios can be generated
TORCS	Racing Simulator, Camera stream, agent positions, testing control policies for vehicles
AIRSIM	Camera stream with depth and semantic segmentation, support for drones
GAZEBO (ROS)	Multi-robot physics simulator employed for path planning & vehicle control in complex 2D & 3D maps
SUMO	Macro-scale modelling of traffic in cities motion planning simulators are used
Constellation	NVIDIA DRIVE Constellation TM simulates camera, LIDAR and radar for autonomous driving
MADRaS	Multi-Agent Autonomous Driving Simulator built on top of TORCS
Flow	Multi-Agent Traffic Control Simulator built on top of SUMO
Highway-env	A gym-based environment that provides a simulator for highway based road topologies
Carcraft	Waymo's simulation environment

Table 7.3: Simulators for autonomous driving (Table credit: [130])

In this work, Gazebo, Robot Operating System (ROS) and the Simulink Robotics System Toolbox are used. Gazebo is an open source simulator capable of simulating robotic systems in complex environments. It has a modular design that allows: various physics engines to be used, high-quality graphics, sensor models, and the creation of 3D worlds and graphical interfaces. In addition, one of the main advantages of gazebo when working with AVs is that it is already included in the ROS packages. ROS is a flexible framework for robotics programming. It provides a collection of tools, packages, libraries that aims to simplify the access and processing of sensors data, eases the task of creating complex robot behavior, and generate an appropriate response for the motors and other actuators of the robot. RViz is a 3-D visualizer of the ROS framework used for displaying data or status information. It is therefore possible to visualize a virtual model of a robot and see it move around. ROS and gazebo are very popular for the developments of AVs.

The Robotics System Toolbox is a Simulink toolbox that provides algorithms and hardware connectivity for the development of autonomous robotics applications. This toolbox provides an interface between MATLAB, Simulink and ROS that allows testing algorithms developed on Simulink on a ROS environment.

7.3/ Experimental setup

The proposed Probabilistic Multi-Controller Architecture (P-MCA) and each of the proposed modules have been implemented first using MATLAB and has then been adapted for transfer to Simulink to meet the specifications of this Ph.D. This first adaptation facilitates integration under ROS as an existing Simulink toolbox called ROS Toolbox provides an interface connecting Simulink with the Robot Operating System. This allows testing the algorithm developed under Simulink on a ROS environment. By using the toolbox Simulink Coder we are able to generate and execute C/C++ code as a node in the implemented ROS architecture.

The framework has been first validated on simulations with the ROS [1] framework and

[1] ROS: https://www.ros.org/

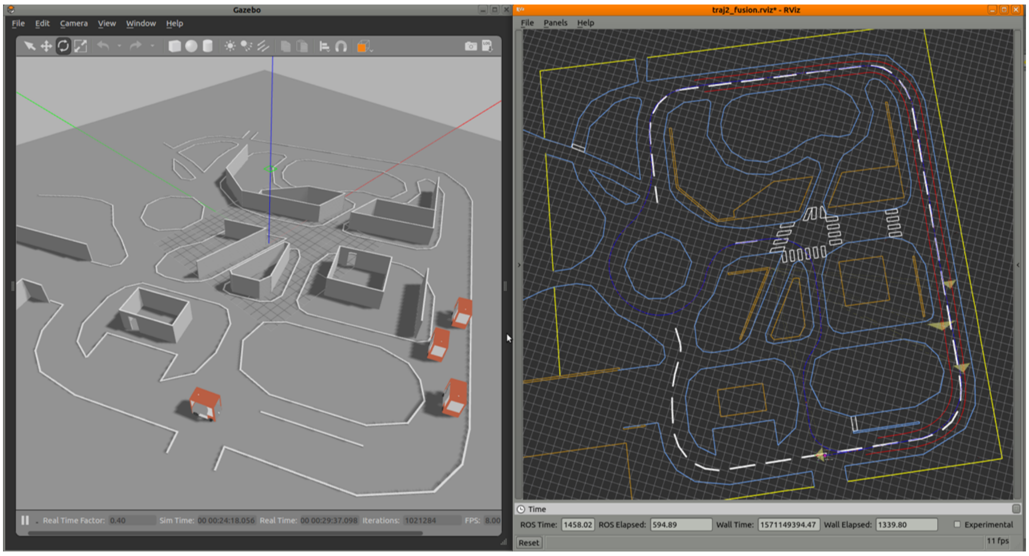


Figure 7.3: ROS (RViz) and Gazebo simulation environment

Gazebo 3D robotics simulator engine [2]. The simulation environment shown in Figure 7.3 provides a realistic physical model for the vehicle and actuators as well as a PAVIN map modeled on Gazebo and on RViz.

Once the first experiments in the simulated environment have been conclusive, they were ported to real vehicles (cf. the procedure shown in Figure 7.4). When Simulink is running

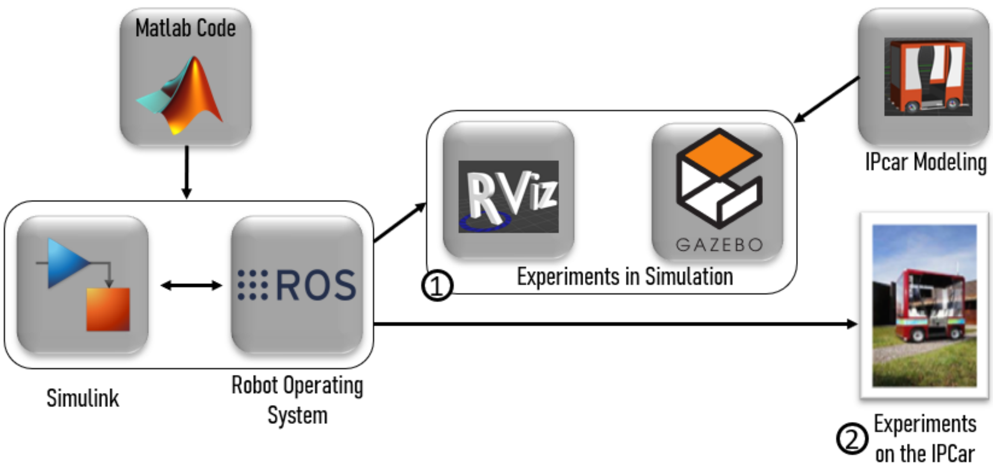


Figure 7.4: Procedure for experimental validations

with ROS, the Simulink architecture is seen as a node. The Robotics System Toolbox provides tools to exchange messages between nodes. In this way, it is possible to retrieve the vehicle positions and to output the speeds, wheel angles and to display the information necessary to check the operations on RViz. The whole process of decision making and choice of maneuver has been developped during my Ph.D. The control sequence to be applied contains only the speed and the steering angle which is compliant with the low level of the IPcar. The Simulink node is responsible of managing all the vehicles in the

[2]Gazebo: http://gazebosim.org/

environment.

Once the Simulink-ROS package is up and running, vehicles can be placed on the RViz/Gazebo map to create scenarios.

7.4/ PERFORMED EXPERIMENTS

The experiments were performed progressively, from a simple overtaking maneuver of a single obstacle vehicle stopped in the ego lane to navigation with multiple dynamic vehicles in the environment. For each use-case, we validated first the well functioning of the scenario on the simulation environments then we proceed with the real-vehicle experiments on the field track. For the real experiments, instead of having the Gazebo simulator it has been replaced by the low level part of an IPcar.

The first tests validated the correct functioning of trajectory tracking and overtaking. Then, the decision-making and the PMCA was validated in the latest tests.

To summarize, the following tests have been performed:

- Trajectory tracking and overtaking of an Obstacle-IPcar at standstill (cf. Figure 7.5).
- Navigation with a dynamic Obstacle-IPcar in the same lane as the Ego-IPcar.
- Navigation with a slower dynamic Obstacle-IPcar in the ego-lane and a static Obstacle-IPcar vehicle on the left lane (cf. Figure 7.6).
- Navigation with two dynamic Obstacles IPcar, one in each lane in front of the Ego-IPcar (cf. Figure 7.8 and 7.9).

For the needs of our experiments, it was necessary that the Ego-IPcar knew important perceptive information such as: its pose in the environment (x, y, θ), the state $(x_{obs}, y_{obs}, \theta_{obs}, v_{obs})$ of the surrounding perceivable Obstacles-IPcar and the lanes.
Indeed, even if the V2V (vehicle-to-vehicle) communication, perception and localization are outside the immediate scope of this work, some solutions have been proposed to successfully conduct this experiments. Each vehicle estimates its current pose using the fusion between RTK-GPS and IMU at a sample time of $T_s = 0.1s$ and can be controlled using the on-board computer.
For safety reasons it was chosen to constrain the velocity on PAVIN platform of the IP-cars to a maximum velocity of $2m/s$. The transmission of localization information between the Obstacles-IPcar and the Ego-IPcar is performed through Wi-Fi communication and the center-line of the lanes are pre-registered. In the prospect of future deployment of the proposed probabilistic framework, it is essential to improve the localization and perception features and propose an end-to-end solution.

7.5/ RESULTS

The first result shown in this section corresponds to a simple lane change maneuver over a static obstacle (cf. Figure 7.5). The obstacle transmits its pose via WiFi communication and thus allow the ego-vehicle to compute the suitable decision at this time. The overall ego-vehicle trajectory is shown in the left part of the figure and illustrates a complete

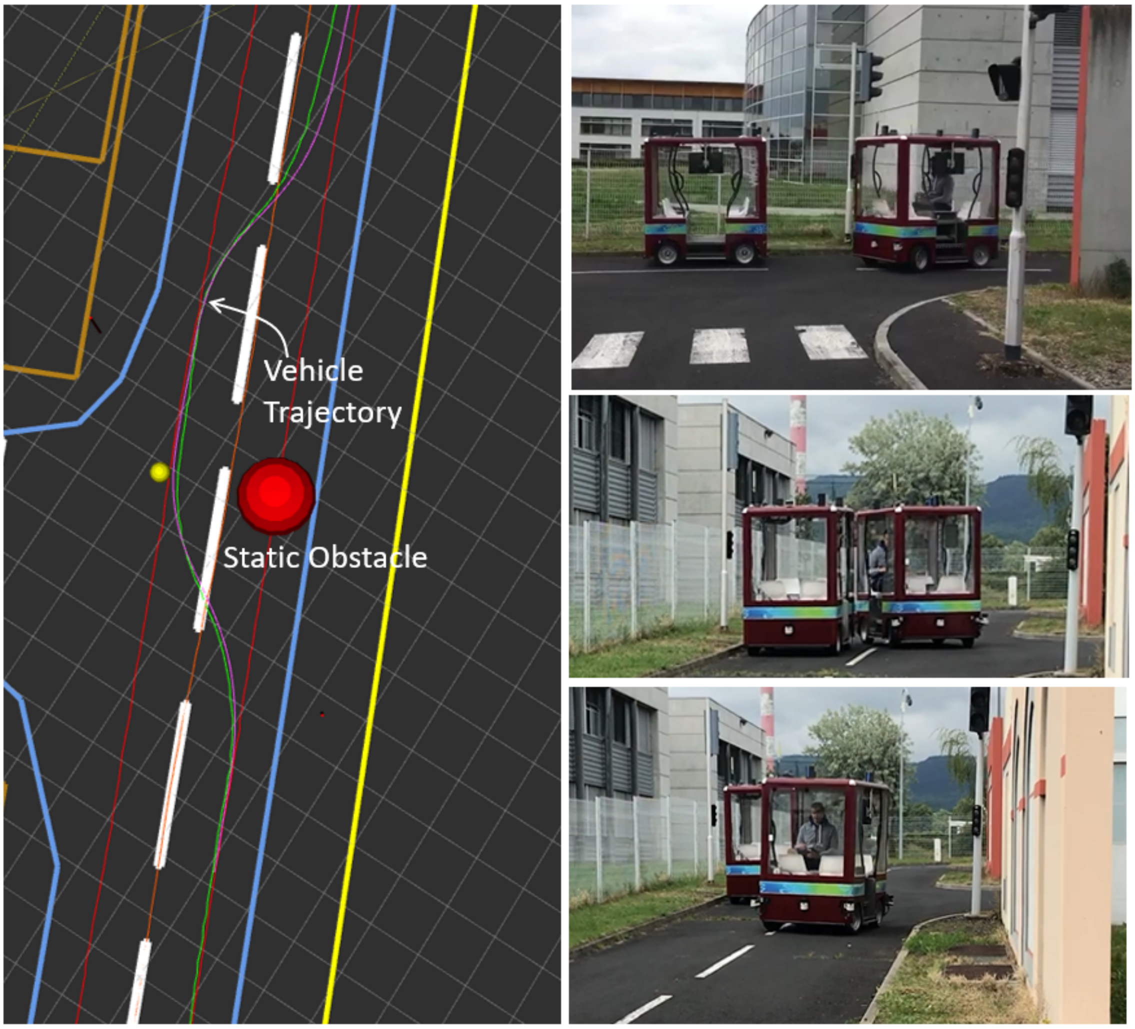


Figure 7.5: Overtaking maneuver of a static Obstacle-IPcar in PAVIN

overtaking maneuver. It starts with a change from the right to the left lane and ends with a change from the left to the right lane.

The second experiment shown in Figure 7.6 highlights how the P-MCA handles the case where a static hindering obstacle is in the left lane while having a slow obstacle in the right lane. In this case the ego-vehicle is forced to stay behind the slow vehicle in the right lane until the static obstacle is over passed and the Ego-vehicle is able to perform the lane change maneuver. For logistic reasons regarding the availability of the IPcars, the static Obstacle-Vehicle was replaced with a virtual one and is highlighted in the figures 7.6 by a static object. The sequencing of decisions is highlighted in Figure 7.7. $Decision_3$ is not active at this stage. We can notice in the figure an oscillatory switching between decisions around the second 35. This is due to discretization. As most real-world problems involve the use of continuous quantities such as mass and temperature and by definition continuous variables have an infinity of possible values which makes specifying conditional probabilities for each value impossible. The common way to handle continuous variable in BN is by using discretization which is dividing possible values into a fixed set of intervals. This has been used in the Maneuver Decision Level (MDL) (cf. Chapter 6 and Figure 6.2) where the values of the Extended Time-To-Collision (ETTC)

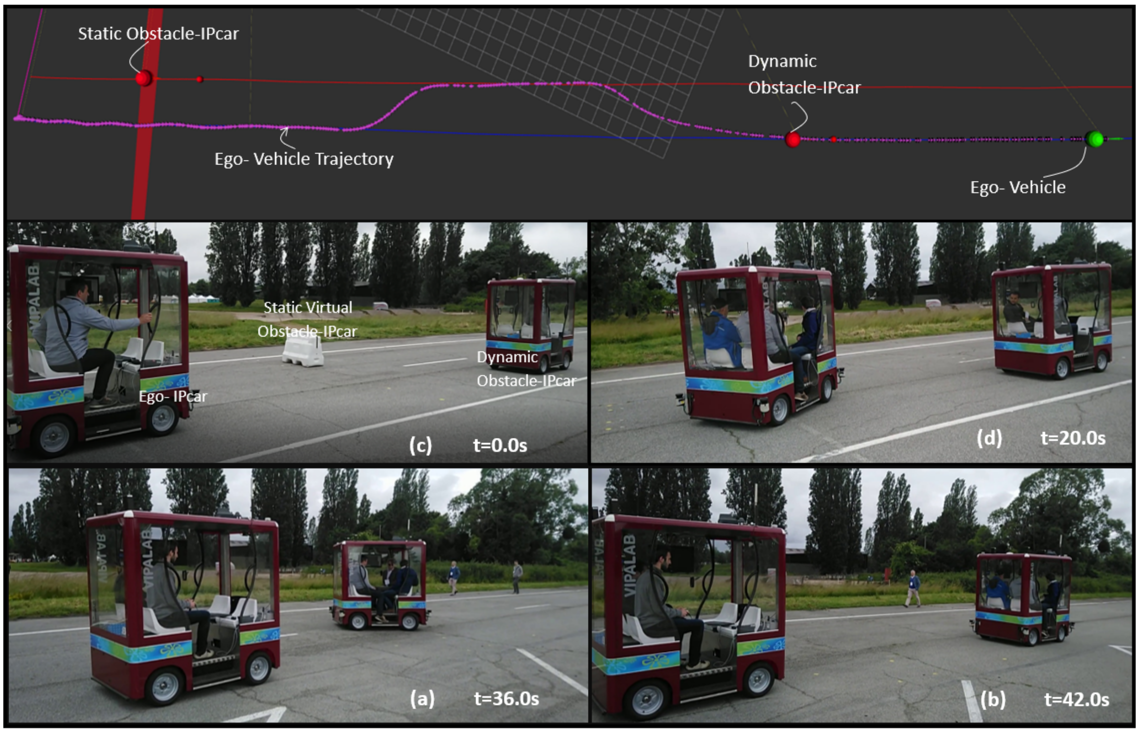
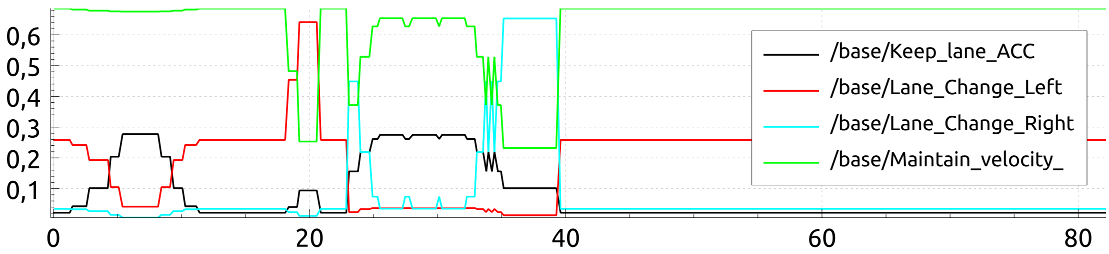
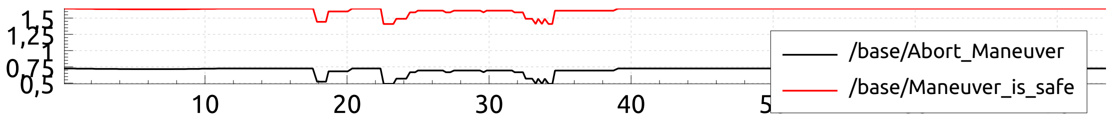


Figure 7.6: Experiment performed during the IEEE Intelligent Vehicles Symposium (Paris, June 2019).



(a) Decision 1

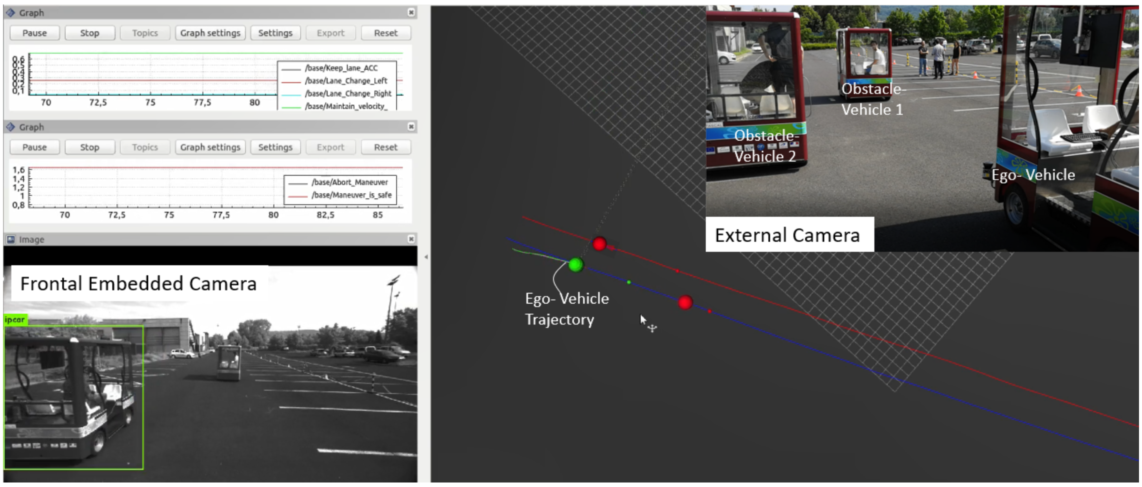


(b) Decision 2

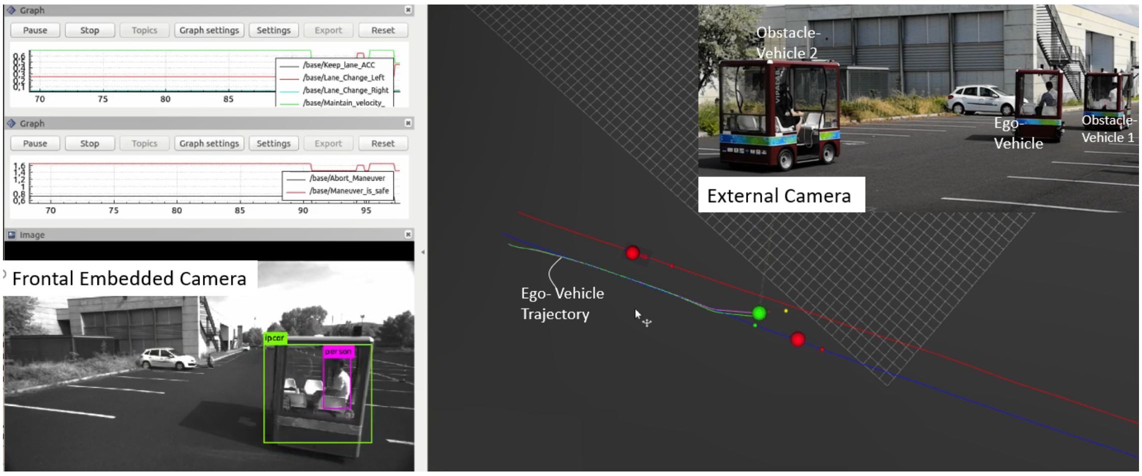
Figure 7.7: Sequencing of decisions corresponding to the experiment shown in Figure 7.6

have been split into a five interval urgency rating. However, discretization often results in a considerable loss of accuracy, very large CPTs and could induces unstable switches between states. In this case, considering a continuous function that allows to update the probability of the node of the status of the lane will allow to smooth the transition between states.

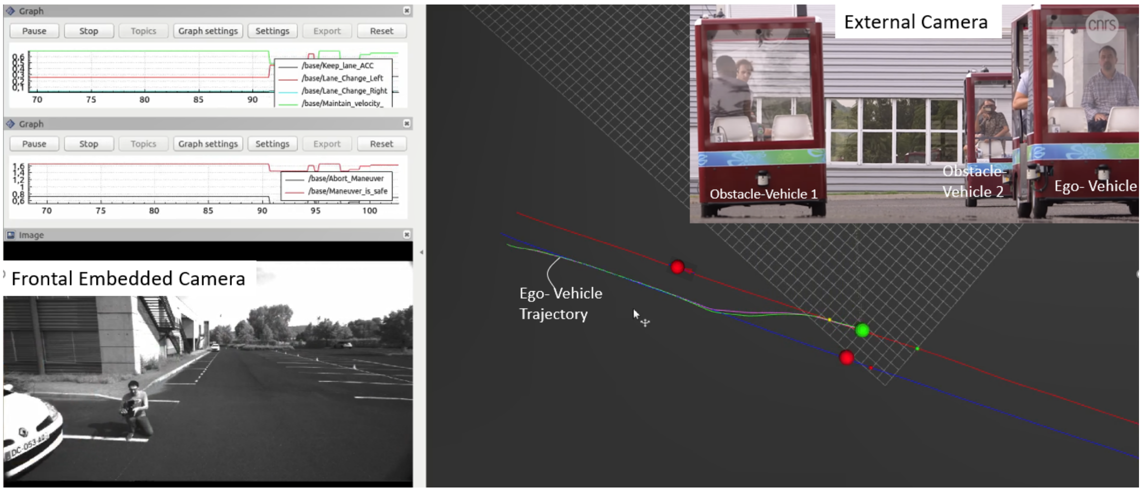
The final experiment have been performed while using three dynamic IPcars. One is con-



(a) Step 1: Lane Keeping in the right lane while maintaining the velocity at t=6.0



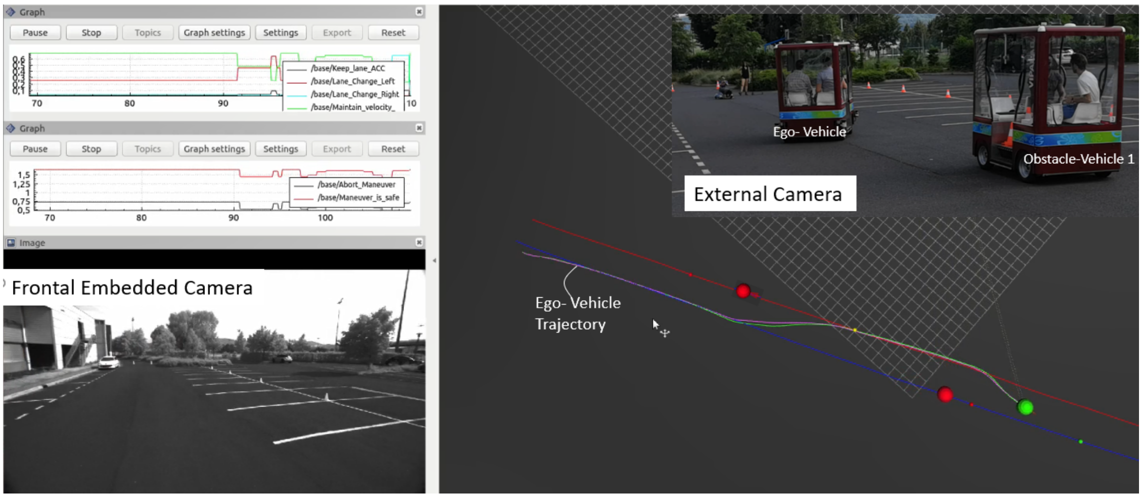
(b) Step 2: Lane Changing to the Left at t=14.0s



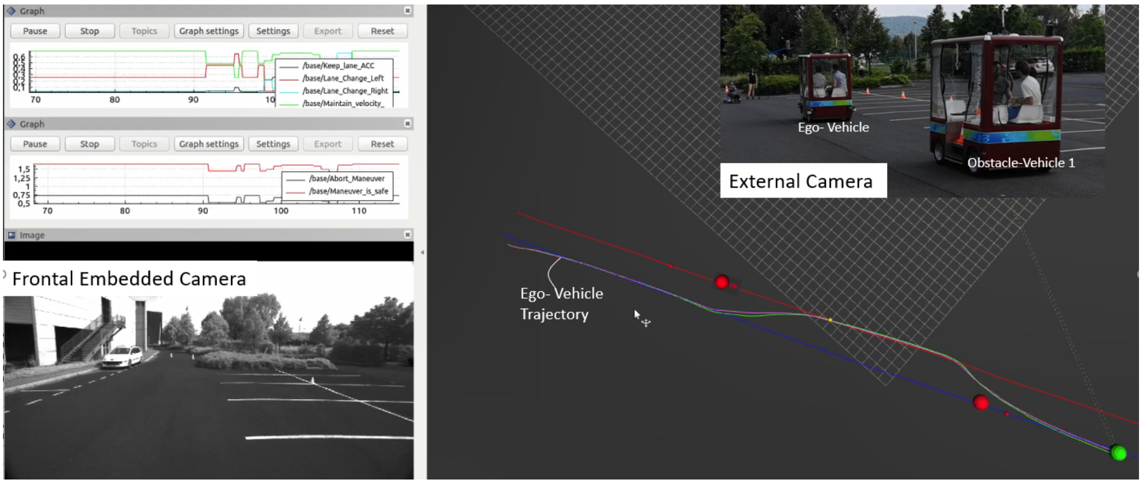
(c) Step 3: Lane Keeping in the left lane while maintaining the velocity at t=18.0s

Figure 7.8: Illustration of the decision-making process: Phase 1 of the overtaking

sidered as the controlled Ego-IPcar and the others are the Obstacle-IPcars. Figure 7.8 and Figure 7.9 show how the P-MCA handles the overtaking maneuver while assessing



(a) Step 1: Lane changing to the right lane at t=25.0s



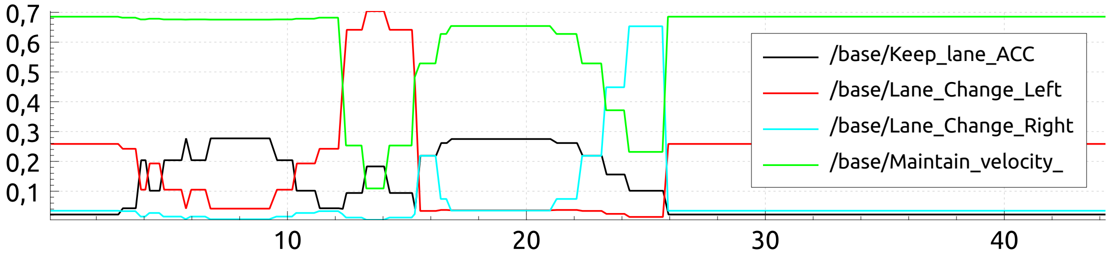
(b) Lane Keeping in the right lane while maintaining the velocity at t=30.0s

Figure 7.9: Illustration of the decision-making process: Phase 2 of the overtaking

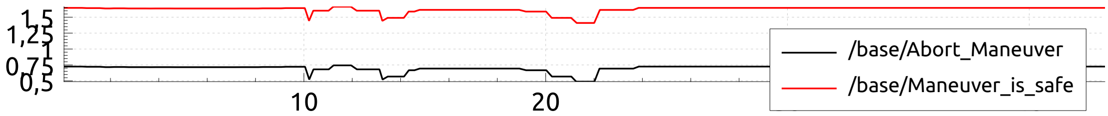
and managing the risk of the achieved task. The Ego-IPcar manage its way through the environment overtake a slow ahead-vehicle (cf. Figure 7.8) and comebacks to its original lane (cf. Figure 7.9). The corresponding consecutive decisions are shown in Figure 7.10.

During the Ph.D., we had also the objective to put all the proposed algorithmic solutions, in a demonstration at the IEEE Intelligent Vehicle Symposium 2019 to illustrate the publications that have been made on the subject. This was followed with multiple cultural events during the year among which two short documentaries for the CNRS 80 years Anniversary https://t.ly/O4ME and for France 3 TV Auvergne https://t.ly/rOhB.

It is to be noted that the different experimentation stated above are available through this link : https://shorturl.at/buQZ8.



(a) Decision 1



(b) Decision 2

Figure 7.10: Sequencing of decisions of the experiments shown in Figures 7.8 and 7.9.

7.6/ Conclusion

In this chapter, the Probabilistic Multi-Controller Architecture (P-MCA) is validated on experimental platforms. For the different experimentation, simulations and contribution shown throughout this dissertation, videos are available through this link: https://shorturl.at/oqEMQ. The strategy in this work was to conduct from one side pure simulations to validate the overall architecture in nominal and emergency situations (presented in sections 5.2.3 and 6.5). And on the other side and in this chapter, was to validate the online functioning of the algorithm for maneuver management as well as its robustness to imperfect input data in real situations.

The proposed Probabilistic Multi-Controller Architecture (P-MCA) and each of the proposed modules have been implemented using MATLAB and Simulink to meet the specifications of this Ph.D. This facilitates integration under ROS as an existing Simulink toolbox called ROS Toolbox provides an interface connecting Simulink with the Robot Operating System. This allows testing the algorithm developed under Simulink on a ROS environment. By using the toolbox Simulink Coder we are able to generate and execute C/C++ code as a part of the implemented ROS architecture and be able to test the P-MCA on the experimental cars. The experiments were performed progressively, from a simple overtaking maneuver of a single obstacle vehicle stopped in the ego lane to navigation with multiple dynamic vehicles in the environment and are shown in this chapter. Even if the simulations and experiments shown in this work have validated several aspects of the proposed P-MCA, it remains several aspects which could improve the validation phase. These are cited in the perspective part of the next chapter.

GENERAL CONCLUSION AND PERSPECTIVES

SUMMARY AND CONCLUSIONS

This dissertation has addressed several aspects for controlling safely and with flexible way autonomous vehicles in uncertain and dynamic environments. An appropriate overall control architecture has been proposed for this purpose, with several levels of decisions and risk assessment and management modules. These kind of components are among the main cornerstone of a robust automotive safety system. More specifically, the topic addressed in this Ph.D. thesis is the full pipeline from path planning, risk assessment and management, until to decision-making and control for autonomous navigation. Overtaking maneuvers in complex and uncertain environment has been chosen as the main addressed use-case. It is summarized in what follows the main elements shown in each of the chapters constituting this manuscript.

The context and motivation of our research works has been described in Chapter 1. The challenge is to propose a system architecture for safe autonomous driving that is unified, generalized, modular and allow to achieve two main objectives. The first one is to derive appropriate decision maneuver in nominal driving and the second is to improve the safety navigation even in emergency situations.

For this purpose, the state of the art review have been divided in three parts. First of all, the automotive safety systems have been reviewed in Chapter 2. They are categorized according to whether they are classical, end-to-end or hybrid approaches. Classical approaches have power in their ability to generalize and formally verify the overall algorithm and modularity. End-to-end approaches has the strength to be compact and self-optimized as all the modules composing a classical architecture (perception, decision-making, planning, control) are integrated in a single entity. The hybrid approaches have the utility of dealing with the limitations raised from each one of the methods. Second of all, the decision-making and the risk estimation problem have been reviewed in Chapter 3. The most important aspects in a decision-making framework is it's ability to solve any situation it falls into, consider uncertainty and unexpected situations while finding the right balance between accuracy and computational expenses. In this regard, probabilistic approaches come out to be the best regarding the defined characteristics as it has the potential to consider the nature of the stochastic dynamics of a traffic environment, they are also able to account for uncertainties through well-known probabilistic algorithms and consider for present and future interactions between interacting entities. Concerning the risk assessment and management strategies, we came out to the conclusion that using multiple complementary criteria can allow to have redundancy in the assessment and improve the accuracy of the acquired information. These strategies allow also to verify the safety of self driving cars online which is mandatory since every traffic situation is unique and a quick response is needed to deal with any emergency situation.

The proposed approach has been thereafter detailed in chapters 5 and 6.
In Chapter 5, it is presented the design of a Probabilistic Multi-Controller Architecture (P-MCA) for safe automated driving under uncertainties. This architecture assembles several interconnected modules that are responsible for assessing the overall surrounding environment by evaluating the collision risk with all observed vehicles considering predictions of road user trajectories, planning driving maneuvers, making the decision on the most suitable actions and investigating the possibility to decide evasive actions if required. The path planning through the used elementary controllers is then presented and is highlighted an adaptation of previous work [214, 216, 217] in order to cope with road-way navigation constraints. Indeed, the proposed Probabilistic Multi-Controller Architecture (P-MCA) is the immediate extension of an already developed Multi-Controller Architecture (MCA) for the navigation of mono- and multi-vehicle navigation in cluttered environments. The proposed controllers are: the Adaptive Cruise Control (ACC), the Lane Keeping Assist (LKA) and the Automatic Lane Changing (ALC). Each vehicle's controller is constituted by a dedicated homogeneous dynamic set-points defined by a pose (x_T, y_T, θ_T) and a velocity v_T. It is highlighted that this set-point formulation is generic enough to define an important number of the vehicle's behaviors. An emphasize is made about the fact that the navigation is performed while tracking these set-points and not in a trajectory following behavior. At each sample time, one of the mentioned controllers is activated and the dynamic set-points are generated. The LKA and ACC controllers uses Frenet reference frame for the dynamic target set-points extraction (cf. subsection 5.2.2.1). The ALC on the other hand is based on Elliptic Limit-Cycle trajectory adapted to road-way navigation and it uses a heuristic to extract its set-points (cf. subsection 5.2.2.2). Once the set-points are defined, a unique stable control law is defined in section 5.1.5 shared by the defined controller will be used to stabilize the errors to zero and thus allow us to reach/track these assigned set-points.
In the same chapter, it is introduced the proposed approach for risk management. Our solution is based on a dual-safety stage strategy. The first stage analyzes the actual driving situation and predicts potential collisions while using a collision based metric the Extended Time-To-Collision (ETTC). This is performed while taking into consideration several dynamic constraints and traffic conditions that are known at the time of planning. The second stage is applied in real-time, during the maneuver achievement, where a safety verification mechanism is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver. This performed while using a novel metric called the Dynamic Predicted Inter-Distance Profile (D-PIDP) between vehicles. It is used to define appropriate analytical formulation for anomaly detection criteria.

In Chapter 6, a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) is proposed as a decision-making framework that is modeled as a sequencing of decisions that a self driving vehicle should take. It corresponds to the main module of the Probabilistic Multi-Controller Architecture (P-MCA). This module is designed to manage several road-way maneuvers under uncertainties. It utilizes the defined safety stages assessment in order to propose discrete actions that allows to: derive appropriate maneuver in a given traffic situation, provide a safety retrospection and verification over the current maneuver risk and take appropriate evasive action autonomously from moving obstacles. In the latter case, it is proposed to compute the corresponding low-level control based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) that allows the ego-vehicle to pursue the advised collision-free evasive trajectory to avert an accident and to guarantee safety at any time.

The reliability and the flexibility of the overall proposed P-MCA and its elementary components have been intensively validated, first in simulated traffic conditions, with various driving scenarios, and secondly, in real-time with the autonomous vehicles available at Institut Pascal on different experimental platform.

PERSPECTIVES

The work presented in this dissertation have shown encouraging results regarding the overall proposed Probabilistic Multi-Controller Architecture (P-MCA) for safe and flexible autonomous navigation. This will allows to explore and extend several research items in the field related to AV. The most important future works are summarized in what follows.

Improve the motion prediction algorithm The used anomaly criteria and the evasive strategy is based on the prediction of all the surrounding vehicles trajectories. So the trajectory prediction of the ego-Vehicle and the surrounding vehicles is a crucial subject. In this dissertation, the ego-vehicle prediction is a maneuver-based trajectory prediction where the predicted trajectory is adapted to the on going maneuver over a longer prediction horizon. The obstacles predicted trajectories in the otherwise is assumed to have a linear motion and constant velocity. However, this is not always realistic. A possibility to improve the quality of our state and maneuver estimations would be to apply an Interaction-aware formalism (cf. section 3.2.3) where instead of relying only on the system dynamics or the on going maneuver, the complex interactions between traffic participants is modeled to estimate and interpret the behaviors and trajectories of traffic participants. A very interesting example is shown in Figure 7.11 where an autonomous Waymo vehicle is driving and have to handle two bicycles (in red) at the rear and a parked car (in green). The algorithm in this case correctly anticipate that the bicycles will interact with parked car and go around it and therefore the Waymo vehicle slows down and let them pass.

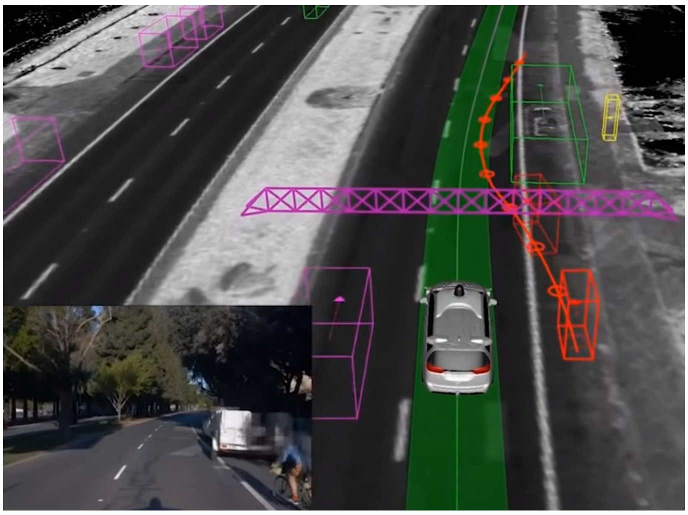


Figure 7.11: Agents in the scene interacting [Image credit : Drago Anguelov (Waymo) - MIT available at https://shorturl.at/nzA78]

A review of the most used algorithms for trajectory prediction have been detailed in the dissertation in Chapter 3, section 3.2.

Driver behavior modeling could be used to encode the preferences of a drive with regards to predefined parameters in order to have a higher quality predictions.

Analytic formalization of Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) Discretization is used in this work to handle the continuous variables (such as the Extended Time-To-Collision (ETTC) in the Maneuver Decision Level (MDL) in Figure 6.3). However, discretization often results in a considerable loss of accuracy, very large Conditional Probability Table (CPT)s and could induces unstable switches between states. In this case, considering a continuous function that updates the probability of the nodes will allow smooth transitions between states.
Another promising alternative is to use a Dynamic Bayesian Network that have the ability to model the system as a series of snapshots or time slices each of which contains a set of random variables. In this way, the information of a node at time t_{k-1} is available at time t_k.

Another area of improvement would be learning the decision maker's utility function. The utility function represents the cost of the decision given the observations. This task can be performed by using driver models of different behaviors and by including information's on the scene or on the criticality of the situation.

Improve the evasive strategy The proposed methodology can also be extended for cases where obstacle-vehicles change lane abruptly in a configuration where a collision might occurs. This obviously, dependent on the surrounding vehicles trajectory predictions.
In any other case, where the optimization is not able to reach a solution (the lateral and longitudinal constraints would be violated) and a collision cannot be avoided, collision mitigation techniques [3] should be applied. These kind of scenarios can be interesting to investigate in future works.
Moreover, the evasive strategy based on Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a proof of concept about the use of optimal profiles to follow, that needs to be refined and optimized particularly regarding the computation time and the prediction horizon.

Extend the contributions to other scene representations Some uses cases have been defined hereafter that could be interesting to investigate in order to validate the genericity of the proposed architecture:

- Merging scenarios: if the road narrows or to deal with highway entrance ramp or exists.
- Urban environment: we can start with simple scenario like lane changing in a two-way traffic.
- Roundabout and intersections.

[3]Actions in order to reduce at maximum the injures of the passengers, and/or the others outside the ego-vehicle [98].

Experimentation and extensive testing Even if the simulations and experiments shown in this work have validated several aspects of the proposed P-MCA, it remains several aspects which could improve the validation phase. Among them, let us cite: performing a quantitative analysis (bash simulation) of the proposed P-MCA. Such a study could be conducted in simulation, using a dedicated simulator (Car Maker or Carla, cf. Table 7.3) to generate realistic scenarios and perception data to be able to draw a statistical analysis. The algorithm's performance could be evaluated in more complex scenarios. The evaluations presented in this chapter involved a maximum of three vehicles, but in theory the algorithm can be used with an arbitrary number of vehicles. Similarly, a two lane layouts were tested in this work but the algorithm can in theory be extended to any number of lane. In the future, experiments should be performed with more vehicles and more lanes to prove the generality of our approach. In addition, the experiments will be extended to the overall P-MCA particularly in emergency situations. These aspects were included in near future perspective of this work.

III

APPENDICES

A

Bayesian Networks (BNs)

In this appendix, a summary of the main elements of axioms of probability theory, Bayesian Network and Decision Networks are given mainly using the following works [163, 181, 201]

A.1/ Probability properties

A probability is a measure over a set of events that satisfies three axioms:

- The measure of each event is between 0 and 1 as $0 \leq P(X = x_i) \leq 1$, where X is a random variable representing an event and x_i are the possible values of X. In general, random variables are denoted by uppercase letters and their values by lowercase letters.
- The measure of the whole set is 1 i.e., $\sum_{i=1}^{n} P(X = x_i) = 1$.
- The probability of a union of disjoint (independent) events is the sum of the probabilities of the individual events i.e., $P(X = x_1 \vee X = x_2) = P(X = x_1) + P(X = x_2)$, where x_1 and x_2 are disjoint. In other cases,

$$\Pr(X \vee Y) = \Pr(X) + \Pr(Y) - \Pr(X \wedge Y)$$

.

A.1.1/ Independence

Independence between variables X and Y can be written as follows:

$$\mathbf{P}(X|Y) = \mathbf{P}(X) \quad \text{or} \quad \mathbf{P}(Y|X) = \mathbf{P}(Y) \quad \text{or} \quad \mathbf{P}(X, Y) = \mathbf{P}(X)\mathbf{P}(Y) \tag{A.1}$$

A.1.2/ Joint probability

The joint probability for two random variables X and Y is defined as fellows:

$$\mathrm{P(X, Y) = P(X)P(Y)} \tag{A.2}$$

A.1.3/ Conditional probability

Mathematically speaking, for any random variables X and Y conditional probabilities are defined as follows:

$$\mathrm{P(X|Y)} = \frac{\mathrm{P(X, Y)}}{\mathrm{P(Y)}} \tag{A.3}$$

This definition of conditional probability can be written in a different form called the **product rule**:

$$P(X \wedge Y) = P(X|Y)P(Y) \tag{A.4}$$

A.1.4/ Marginal probability

One particularly common task is to extract the distribution over some subset of variables or a single variable. This process is called marginalization, or summing out—because we sum up the probabilities for each possible value of the other variables, thereby taking them out of the equation. We can write the following general marginalization rule for any sets of variables $\mathbf{Y}$ and $\mathbf{Z}$:

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z}\in\mathbf{Z}} \mathbf{P}(\mathbf{Y}, \mathbf{z}) \tag{A.5}$$

Where $\sum_{\mathbf{z}\in\mathbf{Z}}$ express the sum over all the possible combinations of values of the set of variables **Z**. A variant of this rule involves conditional probabilities instead of joint probabilities, using the product rule:

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{z})P(\mathbf{z}) \tag{A.6}$$

This rule is called conditioning. Marginalization and conditioning turn out to be useful rules for all kinds of derivations involving probability expressions.

A.1.5/ Bayes' Rule

The more general case of Bayes' rule for multi-valued variables can be written as follows:

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)} \tag{A.7}$$

On the surface, Bayes' rule does not seem very useful. It allows us to compute the single term $\mathbf{P}(Y|X)$ in terms of three parts: $\mathbf{P}(X|Y)$, $\mathbf{P}(Y)$ and $\mathbf{P}(X)$. That seems like two steps backwards, but Bayes' rule is useful in practice because there are many cases where we do have good probability estimates for these three numbers and need to compute the fourth. Often, we perceive as evidence the *effect* of some unknown *cause* and we would like to determine that cause. In that case, Bayes' rule becomes:

$$\mathbf{P}(\,\textit{cause} \mid \textit{effect}\,) = \frac{\mathbf{P}(\,\textit{effect} \mid \textit{cause}\,)\mathbf{P}(\,\textit{cause}\,)}{\mathbf{P}(\,\textit{effect}\,)} \tag{A.8}$$

The conditional probability $\mathbf{P}(\,\textit{effect} \mid \textit{cause}\,)$ quantifies the relationship in the causal direction, whereas $\mathbf{P}(\,\textit{cause} \mid \textit{effect}\,)$ describes the diagnostic direction. In a task such as medical diagnosis, we often have conditional probabilities on causal relationships that is, the doctor knows $\mathbf{P}(\,\textit{symptoms} \mid \textit{disease}\,)$ and want to derive a diagnosis,

$\mathbf{P}$(*disease* | *symptoms*).
Bayes' rule is also useful dealing with combining independent evidence. Imagine having one *cause* for several *effect*. Thus, the full joint distribution can be written as:

$$\mathbf{P}(\textit{ Cause },\ \textit{Effect}_1, \ldots,\ \textit{Effect}_n) = \mathbf{P}(\textit{ Cause}) \prod_i \mathbf{P}(\textit{Efect}_i | \textit{ Cause}) \tag{A.9}$$

A.2/ Bayesian Networks

A Bayesian network is a Directed Acyclic Graph (DAG) in which the quantitative probability information is annotated to each node. The full specification is:

- Each node is a random variable, which could be discrete or continuous.
- A set of links or arrows are directed to connect pairs of nodes. If the arrow is from the node X to node Y, it is said that X is Y's parent. The intuitive meaning of an arrow is typically that X has a direct influence on Y.
- Each node X_i has a conditional probability distribution $P(X_i|Parents(X_i))$ that quantifies the effect of the parents on the node

Viewed as a piece of "syntax", a Bayesian network is a DAG with some numeric parameters attached to each node. These parameters are the conditional probabilities of a node X_i noted as: $\mathbf{P}(X_i|\text{Parents}(X_i))$. The aforementioned conditional probabilities for each node in the network are summarized in a Conditional Probability Table (CPT). One way of defining what the network is about is to describe how it reflects a specific joint distribution across the entire Variables defined as fellows::

$$\mathbf{P}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \mathbf{P}(x_i|\text{parents}(X_i)) \tag{A.10}$$

This equation governs what a Bayesian Network means. Using this equation and the other presented previous allows us to calculate all the probability of each node presented in the Network.

A.2.1/ Decision Network

Decision networks (or Influence diagram) [106] combines Bayesian networks with additional node types for actions and utilities. Decision Networks (DN) allows us to support probabilistic reasoning, decision-making under uncertainty for a given system and yield the capacity to incorporate multiple decision criteria. A DN has three types of nodes:

- Chance nodes U_C represents the set of random variables and their conditional probabilistic dependencies, just as what we can find in Bayesian networks. That can be discrete or continuous nodes.
- Decision nodes U_D represents the choice of action among alternatives. In a DN with multiple decisions [163] (commonly called Multiple Level Decision network (MLDN)), the decision nodes have a temporal order $(D_1, \cdots, D_n)$ which means the action chosen for decision D_{n-1} is part of the information available at decision D_n along with

past observations. This is possible due to the no-forgetting assumption modeled by the link between (D_{n-1} and D_n) that implies perfect recall of past observations and decisions and temporal order of the multiple decisions.

- Utility nodes U_V defines the cost related to the decision. In the literature [181], a normalized utility scale interval $[0, 1]$ is usually used, to compare between some complex scenarios. Both chance and decision nodes can be parents of a utility node, and their state directly affects the utility value.

It is to be noted that the chance nodes are partitioned according to when they are observed.
ł_0 represents the nodes prior to any decision and ł_i is the set of chance nodes observed after decision D_i is taken. A temporal ordering of the MLDN is then: $\text{ł}_0 \prec D_1 \prec \cdots \prec \text{ł}_i \prec D_i$. A DN is then described by a representation of a joint probability distribution:

$$\mathbf{P}(U_C|U_D) = \prod_{X \in U_C} \mathbf{P}(X|parents(X)) \tag{A.11}$$

To identify the most suitable decision, we compute the Expected Utility (EU) for each decision state and the final decision is the alternative maximizing this EU. A Multi-Level Decision Network (MLDN) is a representation of a joint expected utility function due to the chain rule:

$$EU(U_D) = \prod_{X \in U_C} \mathbf{P}(X|parent(X)) \sum_{w \in U_V} U(X_{parent(w)}) \tag{A.12}$$

The ultimate goal of a MLDN is deriving the most suitable decisions given the available evidence following the temporal order of the set of decision nodes U_D (the action chosen for decision D_{n-1} is part of the information available at decision D_n).

$$EU(U_D) = \mathbf{P}(U_C|U_D) \sum_{w \in U_V} U(X_{parents(w)}) \tag{A.13}$$

B

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [94] is an optimization algorithm based on a class of algorithms classified as evolutionary algorithm. This Evolution Strategy (ES) is a stochastic search algorithm suited for search problems that minimizes a non-linear objective function. Search steps are taken by stochastic recombination of points reached so far. This is called mutation. The best of a number of new search points will be chosen based on their fitness or objective function value. The mutation is usually done by adding a realization of a random vector that is normally distributed. Dependencies between variables are represented by a covariance matrix. The CMA is the method to update this covariance matrix [92].

CMA-ES is suited for difficult non-linear non-convex black-box optimization problems in

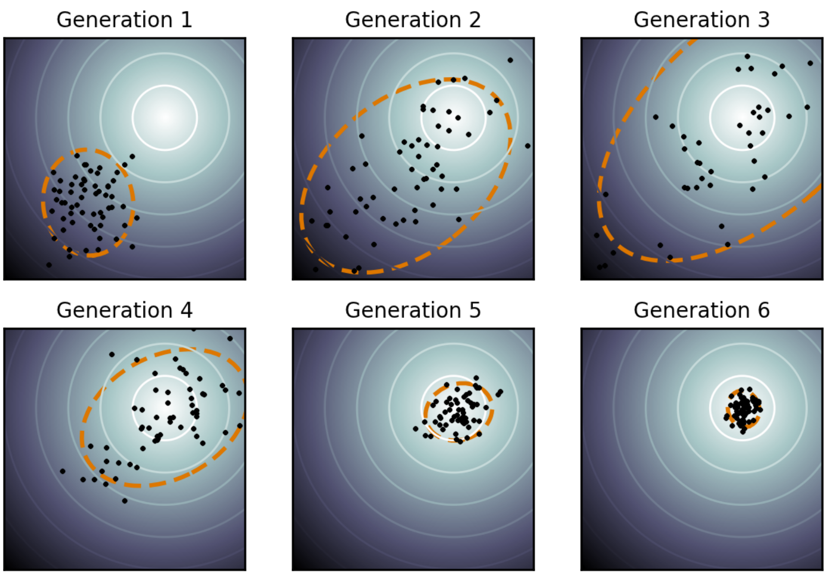


Figure B.1: Illustration of an optimization run on a two-dimensional problem. The spherical optimization landscape is depicted with solid lines. The population (dots) is much larger than necessary, but clearly shows how the distribution of the population changes during the optimization (dotted line). Within a few generations, the population concentrates over the global optimum (Image credit: [225])

continuous domain of which the task of driving, as an activity, also forms a part. Evolutionary algorithms have been found to scan the decision space efficiently, recognizes the most promising solutions and provides new insights into competing sequences. In addition, the likelihood of finding an almost optimum at an early stage of the optimization process is very high. Also, they have great advantages over traditional methods for solving multi-objective optimization problems and they are able to find solutions that are optimal in terms of two or more possibly conflicting objectives. Finally, they provide an effective adaptation of the search distribution to the landscape of the objective function.

For its application the CMA-ES does not require a tedious parameter tuning. In fact, the choice of internal parameters of the strategy is not left to the user except for population size. Finding good strategy parameters is considered as part of the design of the algorithm and not part of its application. The CMA-ES has been applied on various kinds of problem [93, 112, 123].

LIST OF MY PUBLICATIONS

[1] IBERRAKEN, D., ADOUANE, L., AND DENIS, D. **Multi-level bayesian decision-making for safe and flexible autonomous navigation in highway environment**. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct 2018), pp. 3984–3990.

[2] IBERRAKEN, D., ADOUANE, L., AND DENIS, D. **Safe autonomous overtaking maneuver based on inter-vehicular distance prediction and multi-level bayesian decision-making**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (Nov 2018), pp. 3259–3265.

[3] IBERRAKEN, D., ADOUANE, L., AND DENIS, D. **Multi-controller architecture for reliable autonomous vehicle navigation: Combination of model-driven and data-driven formalization**. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (June 2019), pp. 245–251.

[4] IBERRAKEN, D., ADOUANE, L., AND DENIS, D. **Reliable risk management for autonomous vehicles based on sequential bayesian decision networks and dynamic inter-vehicular assessment**. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (June 2019), pp. 2344–2351.

Bibliography

[5] Ackermann, C., Bechtloff, J., et Isermann, R. (2015). **Collision avoidance with combined braking and steering**. In *6th International Munich Chassis Symposium 2015*, pages 199–213. Springer.

[6] Adouane, L. (2009a). **Hybrid and safe control architecture for mobile robot navigation**. In *9th Conference on Autonomous Robot Systems and Competitions, Castelo Branco-Portugal*.

[7] Adouane, L. (2009b). **Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation**. In *Robotica'09, appeared in Portuguese "http://issuu.com/revista-robotica/docs/robotica79" Journal “Robótica”, July 2010*.

[8] Adouane, L. (2016). **Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures**. Taylor & Francis - CRC Press, 228 pages, April 2016. Available: https://www.crcpress.com/Autonomous-Vehicle-Navigation-From-Behavioral-to-Hybrid-Multi-controller/Adouane/978149871558.

[9] Adouane, L. (2017a). **Reactive versus cognitive vehicle navigation based on optimal local and global pelc**. *Robotics and Autonomous Systems (RAS)*, 88:51–70.

[10] Adouane, L. (2017b). **Toward fully autonomous vehicle navigation: From behavioral to hybrid multi-controller architectures**. In *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, pages 85–98.

[11] Adouane, L., Benzerrouk, A., et Martinet, P. (2011). **Mobile robot navigation in cluttered environment using reactive elliptic trajectories**. *IFAC Proceedings Volumes*, 44(1):13801–13806.

[12] Allen, B. L., Shin, B. T., et Cooper, P. J. (1978). **Analysis of traffic conflicts and collisions**. Technical Report.

[13] Altché, F., et de La Fortelle, A. (2017). **An LSTM network for highway trajectory prediction**. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 353–359.

[14] Althoff, M., et Dolan, J. M. (2014). **Online verification of automated road vehicles using reachability analysis**. *IEEE Transactions on Robotics*, 30(4):903–918.

[15] Althoff, M., et Mergel, A. (2011). **Comparison of markov chain abstraction and monte carlo simulation for the safety assessment of autonomous cars**. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1237–1247.

[16] Althoff, M., Stursberg, O., et Buss, M. (2008). **Stochastic reachable sets of interacting traffic participants**. In *2008 IEEE Intelligent Vehicles Symposium*, pages 1086–1092.

[17] Althoff, M., Stursberg, O., et Buss, M. (2009). **Model-based probabilistic collision detection in autonomous driving**. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310.

[18] Ammoun, S., et Nashashibi, F. (2009). **Real time trajectory prediction for collision risk estimation between vehicles**. In *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, pages 417–422.

[19] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., et Mané, D. (2016). **Concrete problems in ai safety**. *arXiv preprint arXiv:1606.06565*.

[20] Aoude, G. S., Luders, B. D., Lee, K. K., Levine, D. S., et How, J. P. (2010). **Threat assessment design for driver assistance system at intersections**. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1855–1862.

[21] Arkin, R. C., Arkin, R. C., et others (1998). **Behavior-based robotics**. MIT press.

[22] Arulkumaran, K., Deisenroth, M. P., Brundage, M., et Bharath, A. A. (2017). **A brief survey of deep reinforcement learning**. *arXiv preprint arXiv:1708.05866*.

[23] Au, T.-C., Fok, C.-L., Vishwanath, S., Julien, C., et Stone, P. (2012). **Evasion planning for autonomous vehicles at intersections**. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1541–1546.

[24] Bae, H., et Kang, Y. (2014). **Decision making methods based on nonlinear model predictive control for emergency collision avoidance in complex situations (iccas 2014)**. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pages 1424–1427.

[25] Bansal, M., Krizhevsky, A., et Ogale, A. (2018). **ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst**. *arXiv preprint arXiv:1812.03079*.

[26] Bechtel, M. G., McEllhiney, E., Kim, M., et Yun, H. (2018). **DeepPicar: A low-cost deep neural network-based autonomous car**. In *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 11–21.

[27] Bellingard, K. (2019). **Intégration algorithmique sous ros et expérimentations sur véhicule réel**. Master's thesis, Polytech' Clermont.

[28] Bellman, R. (1957). **A markovian decision process**. *Journal of mathematics and mechanics*, pages 679–684.

[29] Ben Lakhal, N. M., Adouane, L., Nasri, O., et Hadj Slama, J. B. (2019a). **Risk management for intelligent vehicles based on interval analysis of ttc**. *IFAC-PapersOnLine*, 52(8):338 – 343. 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019.

[30] Ben Lakhal, N. M., Adouane, L., Nasri, O., et Slama, J. B. H. (2019b). **Interval-based solutions for reliable and safe navigation of intelligent autonomous vehicles**. In *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*, pages 124–130.

[31] Ben Lakhal, N. M., Nasri, O., Adouane, L., et Slama, J. B. H. (2020). **Reliable modeling for safe navigation of intelligent vehicles: Analysis of first and second order set-membership ttc**. In *2020 International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 1–8.

[32] Berthelot, A., Tamke, A., Dang, T., et Breuel, G. (2011). **Handling uncertainties in criticality assessment**. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 571–576.

[33] Bishop, C. M. (2013). **Model-based machine learning**. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984).

[34] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et others (2016). **End to end learning for self-driving cars**.

[35] Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., et Muller, U. (2017). **Explaining how a deep neural network trained with end-to-end learning steers a car**.

[36] Borenstein, J., Koren, Y., et others (1991). **The vector field histogram-fast obstacle avoidance for mobile robots**. *IEEE transactions on robotics and automation*, 7(3):278–288.

[37] Bosch (2012). **Chassis systems control driver assistance systems — how much support do german drivers want?** Technical Report, Bosch.

[38] Brannstrom, M., Coelingh, E., et Sjoberg, J. (2010). **Model-based threat assessment for avoiding arbitrary vehicle collisions**. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669.

[39] Brechtel, S. (2015). **Dynamic Decision-making in Continuous Partially Observable Domains: A Novel Method and its Application for Autonomous Driving.** PhD thesis, Karlsruhe Institute of Technology.

[40] Brechtel, S., Gindele, T., et Dillmann, R. (2011). **Probabilistic MDP-behavior planning for cars**. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1537–1542.

[41] Brechtel, S., Gindele, T., et Dillmann, R. (2014). **Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs**. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 392–399.

[42] Broadhurst, A., Baker, S., et Kanade, T. (2005). **Monte carlo road safety reasoning**. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 319–324.

[43] Broggi, A., Cerri, P., Felisa, M., Laghi, M. C., Mazzei, L., et Porta, P. P. (2012). **The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles**. *International Journal of Vehicle Autonomous Systems*, 10(3):147–164.

[44] Brooks, R. (1986). **A robust layered control system for a mobile robot**. *IEEE journal on robotics and automation*, 2(1):14–23.

[45] Buehler, M., Iagnemma, K., et Singh, S. (2009). **The DARPA urban challenge: autonomous vehicles in city traffic**, volume 56. Springer.

[46] Burton, S., Gauerhof, L., et Heinzemann, C. (2017). **Making the case for safety of machine learning in highly automated driving**. In *International Conference on Computer Safety, Reliability, and Security*, pages 5–16. Springer.

[47] Caborni, C., Ko, S. Y., De Momi, E., Ferrigno, G., et y Baena, F. R. (2012). **Risk-based path planning for a steerable flexible probe for neurosurgical intervention**. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 866–871.

[48] Caltagirone, L., Bellone, M., Svensson, L., et Wahde, M. (2017). **Lidar-based driving path generation using fully convolutional neural networks**. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

[49] Cerri, P., Soprani, G., Zani, P., Choi, J., Lee, J., Kim, D., Yi, K., et Broggi, A. (2011). **Computer vision at the hyundai autonomous challenge**. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 777–783.

[50] Chakarov, A., Nori, A., Rajamani, S., Sen, S., et Vijaykeerthy, D. (2016). **Debugging machine learning tasks**. *arXiv preprint arXiv:1603.07292*.

[51] Chang, T., Hsu, C., Wang, C., et Yang, L. (2008). **Onboard measurement and warning module for irregular vehicle behavior**. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):501–513.

[52] Chen, C., Seff, A., Kornhauser, A., et Xiao, J. (2015). **Deepdriving: Learning affordance for direct perception in autonomous driving**. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730.

[53] Chen, Z., et Huang, X. (2017). **End-to-end learning for lane keeping of self-driving cars**. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1856–1860.

[54] Chirca, M. (2016). **Perception pour la navigation et le contrôle des robots mobiles. Application à un système de voiturier autonome**. PhD thesis, Clermont-Ferrand 2.

[55] Claussmann, L., Revilloud, M., Gruyer, D., et Glaser, S. (2019). **A review of motion planning for highway autonomous driving**. *IEEE Transactions on Intelligent Transportation Systems*.

[56] Codevilla, F., Miiller, M., López, A., Koltun, V., et Dosovitskiy, A. (2018). **End-to-end driving via conditional imitation learning**. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9.

[57] Committee, S. O.-R. A. V. S. (2014). **Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems**. *SAE Standard J*, 3016:1–16.

[58] Coskun, S., et Langari, R. (2017). **Enhanced vehicle handling performance for an emergency lane changing controller in highway driving**. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 334–340.

[59] Coué, C., Pradalier, C., Laugier, C., Fraichard, T., et Bessière, P. (2006). **Bayesian occupancy filtering for multitarget tracking: an automotive application**. *The International Journal of Robotics Research*, 25(1):19–30.

[60] Dagli, I., et Reichardt, D. (2002). **Motivation-based approach to behavior prediction**. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, pages 227–233.

[61] De Iaco, R., Smith, S. L., et Czarnecki, K. (2020). **Universally safe swerve manoeuvres for autonomous driving**. *arXiv preprint arXiv:2001.11159*.

[62] De Luca, A., Oriolo, G., et Samson, C. (1998). **Feedback control of a nonholonomic car-like robot**. In *Robot motion planning and control*, pages 171–253. Springer.

[63] Dickmanns, E. D., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Thomanek, F., et Schiehlen, J. (1994). **The seeing passenger car'vamors-p'**. In *Proceedings of the Intelligent Vehicles' 94 Symposium*, pages 68–73.

[64] Dickmanns, E. D., et others (1997). **Vehicles capable of dynamic vision**. In *IJCAI*, volume 97, pages 1577–1592.

[65] Dickmanns, E. D., et Graefe, V. (1988). **Dynamic monocular machine vision**. *Machine vision and applications*, 1(4):223–240.

[66] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., et Koltun, V. (2017). **CARLA: An open urban driving simulator**. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

[67] Drake, A. W. (1962). **Observation of a Markov process through a noisy channel**. PhD thesis, Massachusetts Institute of Technology.

[68] eetimes (2019). A Wave of AV Safety Standards to Hit in 2020, Consulted on July 2020: https://www.eetimes.com/a-wave-of-av-safety-standards-to-hit-in-2020/.

[69] Elfes, A. (1989). **Using occupancy grids for mobile robot perception and navigation**. *Computer*, 22(6):46–57.

[70] Firl, J., et Tran, Q. (2011). **Probabilistic maneuver prediction in traffic scenarios.** In *ECMR*, pages 89–94.

[71] Föllmer, H., Schied, A., et Lyons, T. J. (2004). **Stochastic finance. an introduction in discrete time**. *The Mathematical Intelligencer*, 26(4):67–68.

[72] Forbes, J., Huang, T., Kanazawa, K., et Russell, S. (1995). **The batmobile: Towards a bayesian automated taxi**. In *IJCAI*, volume 95, pages 1878–1885.

[73] Fraichard, T. (2007). **A short paper about motion safety**. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1140–1145.

[74] Fraichard, T., et Asama, H. (2004). **Inevitable collision states—a step towards safer robots?** *Advanced Robotics*, 18(10):1001–1024.

[75] Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., et Herrtwich, R. (2013). **Making bertha see**. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 214–221.

[76] Funke, J., Brown, M., Erlien, S. M., et Gerdes, J. C. (2016). **Collision avoidance and stabilization for autonomous vehicles in emergency scenarios**. *IEEE Transactions on Control Systems Technology*, 25(4):1204–1216.

[77] Gal, Y. (2016). **Uncertainty in deep learning**. *University of Cambridge*, 1:3.

[78] Gat, E., Bonnasso, R. P., Murphy, R., et others (1998). **On three-layer architectures**. *Artificial intelligence and mobile robots*, 195:210.

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

[80] Gerssen, S. G., et Rothkrantz, L. J. (2006). **Bayesian network learning for rare events**. In *Proceedings of the International Conference on Computer Systems and Technologies*.

[81] Gim, S., Adouane, L., Lee, S., et Derutin, J.-P. (2017). **Clothoids composition method for smooth path generation of car-like vehicle navigation**. *Journal of Intelligent & Robotic Systems*, 88(1):129–146.

[82] Gindele, T., Brechtel, S., et Dillmann, R. (2010). **A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments**. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1625–1631.

[83] Gindele, T., Brechtel, S., et Dillmann, R. (2015). **Learning driver behavior models from traffic observations for decision making and planning**. *IEEE Intelligent Transportation Systems Magazine*, 7(1):69–79.

[84] Godthelp, H., Milgram, P., et Blaauw, G. J. (1984). **The development of a time-related measure to describe driving strategy**. *Human factors*, 26(3):257–268.

[85] González, D., Pérez, J., Milanés, V., et Nashashibi, F. (2015). **A review of motion planning techniques for automated vehicles**. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145.

[86] Goodall, N. J. (2014). **Ethical decision making during automated vehicle crashes**. *Transportation Research Record*, 2424(1):58–65.

[87] Greene, D., Liu, J., Reich, J., Hirokawa, Y., Shinagawa, A., Ito, H., et Mikami, T. (2011). **An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists**. *IEEE Transactions on intelligent transportation systems*, 12(4):942–953.

[88] Grigorescu, S., Trasnea, B., Cocias, T., et Macesanu, G. (2019). **A survey of deep learning techniques for autonomous driving**. *Journal of Field Robotics*.

[89] Gu, S., Lillicrap, T., Sutskever, I., et Levine, S. (2016). **Continuous deep q-learning with model-based acceleration**. In *International Conference on Machine Learning*, pages 2829–2838.

[90] Gunning, D. (2017). **Explainable artificial intelligence (xai)**. *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2.

[91] Guo, J., Kurup, U., et Shah, M. (2019). **Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving**. *IEEE Transactions on Intelligent Transportation Systems*.

[92] Hansen, N. (2016). **The CMA evolution strategy: A tutorial**. *arXiv preprint arXiv:1604.00772*.

[93] Hansen, N., Niederberger, A. S., Guzzella, L., et Koumoutsakos, P. (2008). **A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion**. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197.

[94] Hansen, N., et Ostermeier, A. (2001). **Completely derandomized self-adaptation in evolution strategies**. *Evolutionary computation*, 9(2):159–195.

[95] Hayward, J. C. (1972). **Near miss determination through use of a scale of danger**.

[96] Hecker, S., Dai, D., et Van Gool, L. (2018). **End-to-end learning of driving models with surround-view cameras and route planners**. In *Proceedings of the european conference on computer vision (eccv)*, pages 435–453.

[97] Hillel, A. B., Lerner, R., Levi, D., et Raz, G. (2014). **Recent progress in road and lane detection: a survey**. *Machine vision and applications*, 25(3):727–745.

[98] Hillenbrand, J., Spieker, A. M., et Kroschel, K. (2006). **A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs**. *IEEE Transactions on intelligent transportation systems*, 7(4):528–540.

[99] Hilscher, M., Linker, S., et Olderog, E.-R. (2013). **Proving safety of traffic manoeuvres on country roads**. In *Theories of Programming and Formal Methods*, pages 196–212. Springer.

[100] Hirsch, K., Hilgert, J., Lalo, W., Schramm, D., et Hiller, M. (2005). **Optimization of emergency trajectories for autonomous vehicles with respect to linear vehicle dynamics**. In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pages 528–533.

[101] Hoel, C.-J., Wolff, K., et Laine, L. (2018). **Automated speed and lane change decision making using deep reinforcement learning**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2148–2155.

[102] Hou, J., List, G. F., et Guo, X. (2014). **New algorithms for computing the time-to-collision in freeway traffic simulation models**. *Computational intelligence and neuroscience*, 2014:57.

[103] Hou, Z.-S., et Wang, Z. (2013). **From model-based control to data-driven control: Survey, classification and perspective**. *Information Sciences*, 235:3 – 35. Data-based Control, Decision, Scheduling and Fault Diagnostics.

[104] Houenou, A., Bonnifait, P., Cherfaoui, V., et Yao, W. (2013). **Vehicle trajectory prediction based on motion model and maneuver recognition**. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4363–4369.

[105] Howard, R. A. (1960). **Dynamic programming and markov processes.**

[106] Howard, R. A., et Matheson, J. E. (1984). **The principles and applications of decision analysis**. *Strategic Decisions Group, Palo Alto, CA*, pages 719–762.

[107] Hruschka, E. R., et do Carmo Nicoletti, M. (2013). **Roles Played by Bayesian Networks in Machine Learning: An Empirical Investigation**, pages 75–116. Springer Berlin Heidelberg, Berlin, Heidelberg.

[108] Huang, X., Kwiatkowska, M., Wang, S., et Wu, M. (2017). **Safety verification of deep neural networks**. In *International Conference on Computer Aided Verification*, pages 3–29. Springer.

[109] Hubmann, C., Becker, M., Althoff, D., Lenz, D., et Stiller, C. (2017). **Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles**. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1671–1678.

[110] Hupfer, C. (1997). **Deceleration to safety time (dst)-a useful figure to evaluate traffic safety**. In *ICTCT Conference Proceedings of Seminar*, volume 3, pages 5–7.

[111] Häger, D., et Andersen, L. B. (2010). **A knowledge based approach to loss severity assessment in financial institutions using bayesian networks and loss determinants**. *European Journal of Operational Research*, 207(3):1635 – 1644.

[112] Ibáñez, O., Ballerini, L., Cordón, O., Damas, S., et Santamaría, J. (2009). **An experimental study on the applicability of evolutionary algorithms to craniofacial superimposition in forensic identification**. *Information Sciences*, 179(23):3998–4028.

[113] IEEE (2020). **IEEEP2846 norm : "A Formal Model for Safety Considerations in Automated Vehicle Decision Making"**, Consulted on July 2020: https://sagroups.ieee.org/2846/.

[114] INRIA (2013). **The CMA Evolution Strategy**, Consulted on July 2020: http://cma.gforge.inria.fr/cmaesintro.html.

[115] Ionita, S. (2017). **Autonomous vehicles: from paradigms to technology**. In *IOP Conference Series: Materials Science and Engineering*, volume 252. IOP Publishing.

[116] IPDS (2013). **The Institut Pascal Data Sets** Consulted on July 2020: http://ipds.univ-bpclermont.fr.

[117] Jaritz, M., De Charette, R., Toromanoff, M., Perot, E., et Nashashibi, F. (2018). **End-to-end race driving with deep reinforcement learning**. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2070–2075.

[118] Jiménez, F., Naranjo, J. E., et García, F. (2013). **An improved method to calculate the time-to-collision of two vehicles**. *International Journal of Intelligent Transportation Systems Research*, 11(1):34–42.

[119] Kanade, T., Thorpe, C., et Whittaker, W. (1986). **Autonomous land vehicle project at cmu**. In *Proceedings of the 1986 ACM fourteenth annual conference on Computer science*, pages 71–80.

[120] Kasmi, A., Denis, D., Aufrere, R., et Chapuis, R. (2018). **Map matching and lanes number estimation with openstreetmap**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2659–2664.

[121] Katrakazas, C., Quddus, M., et Chen, W.-H. (2019). **A new integrated collision risk assessment methodology for autonomous vehicles**. *Accident Analysis & Prevention*, 127:61–79.

[122] Katrakazas, C., Quddus, M. A., et Chen, W.-H. (2017). **A new methodology for collision risk assessment of autonomous vehicles**. In *Presented at the Transportation Research Board 96th Annual Meeting, January*, volume 8.

[123] Kelemen, A., Abraham, A., et Liang, Y. (2008). **Computational intelligence in medical informatics**, volume 85. Springer Science & Business Media.

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

[125] Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., et Shah, A. (2019). **Learning to drive in a day**. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254.

[126] Khatib, O. (1986). **Real-time obstacle avoidance for manipulators and mobile robots**. In *Autonomous robot vehicles*, pages 396–404. Springer.

[127] Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C., et Choi, J. W. (2017a). **Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network**. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 399–404.

[128] Kim, D.-H., et Kim, J.-H. (2003). **A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer**. *Robotics and Autonomous Systems*, 42(1):17–30.

[129] Kim, J., et Kum, D. (2017b). **Collision risk assessment algorithm via lane-based probabilistic motion prediction of surrounding vehicles**. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2965–2976.

[130] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., et Pérez, P. (2020). **Deep reinforcement learning for autonomous driving: A survey**. *arXiv preprint arXiv:2002.00444*.

[131] Kress-Gazit, H., et Pappas, G. J. (2008). **Automatically synthesizing a planning and control subsystem for the darpa urban challenge**. In *2008 IEEE International Conference on Automation Science and Engineering*, pages 766–771.

[132] Kuderer, M., Gulati, S., et Burgard, W. (2015). **Learning driving styles for autonomous vehicles from demonstration**. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2646.

[133] Laconte, J., Debain, C., Chapuis, R., Pomerleau, F., et Aufrère, R. (2019). **Lambda-field: A continuous counterpart of the bayesian occupancy grid for risk assessment**. *arXiv preprint arXiv:1903.02285*.

[134] Lakhal, N. M. B., Adouane, L., Nasri, O., et Slama, J. B. H. (2019). **Risk management for intelligent vehicles based on interval analysis of ttc**. In *IFAC Symposium on Intelligent Autonomous Vehicles, Gdansk, Poland*, pages 1–6.

[135] Laugier, C., Paromtchik, I. E., Perrollaz, M., Yong, M., Yoder, J.-D., Tay, C., Mekhnacha, K., et Nègre, A. (2011). **Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety**. *IEEE Intelligent Transportation Systems Magazine*, 3(4):4–19.

[136] Laureshyn, A., Svensson, Å., et Hydén, C. (2010). **Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation**. *Accident Analysis & Prevention*, 42(6):1637–1646.

[137] Lee, S. E., Olsen, E. C., Wierwille, W. W., et others (2004). **A comprehensive examination of naturalistic lane-changes**. Technical Report, United States. National Highway Traffic Safety Administration.

[138] Lefèvre, S., Laugier, C., et Ibañez-Guzmán, J. (2012). **Risk assessment at road intersections: Comparing intention and expectation**. In *2012 IEEE Intelligent Vehicles Symposium*, pages 165–171.

[139] Lefèvre, S., Vasquez, D., et Laugier, C. (2014). **A survey on motion prediction and risk assessment for intelligent vehicles**. *ROBOMECH journal*, 1(1):1.

[140] Leonhardt, V., et Wanielik, G. (2018). **Recognition of lane change intentions fusing features of driving situation, driver behavior, and vehicle movement by means of neural networks**. In *Advanced Microsystems for Automotive Applications 2017*, pages 59–69. Springer.

[141] Lethaus, F., Baumann, M. R., Köster, F., et Lemmer, K. (2011). **Using pattern recognition to predict driver intent**. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 140–149. Springer.

[142] Li, J., Dai, B., Li, X., Xu, X., et Liu, D. (2019). **A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification**. *Electronics*, 8(1):40.

[143] Li, L., Ota, K., et Dong, M. (2018). **Humanlike driving: empirical decision-making system for autonomous vehicles**. *IEEE Transactions on Vehicular Technology*, 67(8):6814–6823.

[144] Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A. I., et Martinet, P. (2012). **Map partitioning to approximate an exploration strategy in mobile robotics**. *Multiagent and Grid Systems (MAGS)*, 8(3):275–288.

[145] Lucas, P. (2001). **Bayesian networks in medicine: a model-based approach to medical decision making**.

[146] Magdici, S., et Althoff, M. (2016). **Fail-safe motion planning of autonomous vehicles**. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 452–458.

[147] Majumdar, A., et Pavone, M. (2020). **How should a robot assess risk? towards an axiomatic theory of risk in robotics**. In *Robotics Research*, pages 75–84. Springer.

[148] Mammar, S., Glaser, S., et Netto, M. (2006). **Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting**. *IEEE Transactions on intelligent transportation systems*, 7(2):226–241.

[149] Mandalia, H. M., et Salvucci, M. D. D. (2005). **Using support vector machines for lane-change detection**. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 49, pages 1965–1969. SAGE Publications Sage CA: Los Angeles, CA.

[150] Mazor, E., Averbuch, A., Bar-Shalom, Y., et Dayan, J. (1998). **Interacting multiple model methods in target tracking: a survey**. *IEEE Transactions on aerospace and electronic systems*, 34(1):103–123.

[151] McAllister, R., Gal, Y., Kendall, A., Van Der Wilk, M., Shah, A., Cipolla, R., et Weller, A. (2017). **Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning**. International Joint Conferences on Artificial Intelligence, Inc.

[152] Meyer-Delius, D., Plagemann, C., Von Wichert, G., Feiten, W., Lawitzky, G., et Burgard, W. (2008). **A probabilistic relational model for characterizing situations in dynamic multi-agent systems**. In *Data analysis, machine learning and applications*, pages 269–276. Springer.

[153] Minderhoud, M. M., et Bovy, P. H. (2001). **Extended time-to-collision measures for road traffic safety assessment**. *Accident Analysis & Prevention*, 33(1):89–97.

[154] Mirchevska, B., Pek, C., Werling, M., Althoff, M., et Boedecker, J. (2018). **High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2156–2162.

[155] Mitsch, S., Loos, S. M., et Platzer, A. (2012). **Towards formal verification of freeway traffic control**. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pages 171–180. IEEE Computer Society.

[156] Moras, J., Cherfaoui, V., et Bonnifait, P. (2011). **Credibilist occupancy grids for vehicle perception in dynamic environments**. In *2011 IEEE International Conference on Robotics and Automation*, pages 84–89.

[157] Mouhagir, H., Talj, R., Cherfaoui, V., Guillemard, F., et Aioun, F. (2016). **A markov decision process-based approach for trajectory planning with clothoid tentacles**. In *IEEE Intelligent Vehicles Symposium (IV 2016)*, pages 1254–1259.

[158] Müller, M., Dosovitskiy, A., Ghanem, B., et Koltun, V. (2018). **Driving policy transfer via modularity and abstraction**. *arXiv preprint arXiv:1804.09364*.

[159] Muller, U., Ben, J., Cosatto, E., Flepp, B., et Cun, Y. L. (2006). **Off-road obstacle avoidance through end-to-end learning**. In *Advances in neural information processing systems*, pages 739–746.

[160] Murphy, K. P., et Russell, S. (2002). **Dynamic bayesian networks: representation, inference and learning**.

[161] Naumann, M., Königshof, H., et Stiller, C. (2019). **Provably safe and smooth lane changes in mixed trafic**. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1832–1837.

[162] Ngai, D. C. K., et Yung, N. H. C. (2011). **A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers**. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):509–522.

[163] Nielsen, T. D., et Jensen, F. V. (2009). **Bayesian networks and decision graphs**. Springer Science & Business Media.

[164] Nilsson, J., Silvlin, J., Brannstrom, M., Coelingh, E., et Fredriksson, J. (2016). **If, when, and how to perform lane change maneuvers on highways**. *IEEE Intelligent Transportation Systems Magazine*, 8(4):68–78.

[165] Nilsson, J., et Sjöberg, J. (2013). **Strategic decision making for automated driving on two-lane, one way roads using model predictive control**. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1253–1258.

[166] Noh, S., et An, K. (2017). **Decision-making framework for automated driving in highway environments**. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):58–71.

[167] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., et Frazzoli, E. (2016). **A survey of motion planning and control techniques for self-driving urban vehicles**. *IEEE Transactions on intelligent vehicles*, 1(1):33–55.

[168] Pagac, D., Nebot, E. M., et Durrant-Whyte, H. (1998). **An evidential approach to map-building for autonomous vehicles**. *IEEE Transactions on Robotics and Automation*, 14(4):623–629.

[169] Pan, X., You, Y., Wang, Z., et Lu, C. (2017a). **Virtual to real reinforcement learning for autonomous driving**. *arXiv preprint arXiv:1704.03952*.

[170] Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., et Boots, B. (2017b). **Agile autonomous driving using end-to-end deep imitation learning**. *arXiv preprint arXiv:1709.07174*.

[171] Pearl, J. (2014). **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Elsevier.

[172] Pearl, J. (2018). **Theoretical impediments to machine learning with seven sparks from the causal revolution**. *arXiv preprint arXiv:1801.04016*.

[173] Pek, C., et Althoff, M. (2018a). **Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1447–1454.

[174] Pek, C., et Althoff, M. (2018b). **Efficient computation of invariably safe states for motion planning of self-driving vehicles**. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3523–3530.

[175] Pek, C., Zahn, P., et Althoff, M. (2017). **Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules**. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1477–1483.

[176] Perot, E., Jaritz, M., Toromanoff, M., et De Charette, R. (2017). **End-to-end driving in a realistic racing game with deep reinforcement learning**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 3–4.

[177] Pomerleau, D. A. (1989). **Alvinn: An autonomous land vehicle in a neural network**. In *Advances in neural information processing systems*, pages 305–313.

[178] Rajamani, R. (2011). **Vehicle dynamics and control**. Springer Science & Business Media.

[179] Rosique, F., Navarro, P. J., Fernández, C., et Padilla, A. (2019). **A systematic review of perception system and simulators for autonomous vehicles research**. *Sensors*, 19(3):648.

[180] Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., et Arras, K. O. (2019). **Human motion trajectory prediction: A survey**. *arXiv preprint arXiv:1905.06113*.

[181] Russell, S. J., et Norvig, P. (2002). **Artificial intelligence: a modern approach (2nd edition)**. (Englewood Cliffs: Prentice Hall.

[182] Salas, J., Jiménez, H., González, J., et Hurtado, J. (2007). **Detecting unusual activities at vehicular intersections**. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 864–869.

[183] Samson, C. (1995). **Control of chained systems application to path following and time-varying point-stabilization of mobile robots**. *IEEE Transactions on Automatic Control*, 40(1):64–77.

[184] Sandblom, F., et Brännström, M. (2011). **Probabilistic threat assessment and driver modeling in collision avoidance systems**. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 914–919.

[185] Sankavaram, C., Pattipati, B., Kodali, A., Pattipati, K., Azam, M., Kumar, S., et Pecht, M. (2009). **Model-based and data-driven prognosis of automotive and electronic systems**. In *2009 IEEE International Conference on Automation Science and Engineering*, pages 96–101.

[186] Schubert, R. (2012). **Evaluating the utility of driving: Toward automated decision making under uncertainty**. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):354–364.

[187] Schubert, R., Richter, E., et Wanielik, G. (2008). **Comparison and evaluation of advanced motion models for vehicle tracking**. In *2008 11th International Conference on Information Fusion*, pages 1–6.

[188] Schulz, J., Hubmann, C., Löchner, J., et Burschka, D. (2018a). **Interaction-aware probabilistic behavior prediction in urban environments**. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3999–4006.

[189] Schulz, J., Hubmann, C., Löchner, J., et Burschka, D. (2018b). **Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction**. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1467–1474.

[190] Seshia, S. A., Sadigh, D., et Sastry, S. S. (2016). **Towards verified artificial intelligence**. *arXiv preprint arXiv:1606.08514*.

[191] Shalev-Shwartz, S., Shammah, S., et Shashua, A. (2017). **On a formal model of safe and scalable self-driving cars**. *arXiv preprint arXiv:1708.06374*.

[192] Sheridan, T. B. (2016). **Human–robot interaction: status and challenges**. *Human factors*, 58(4):525–532.

[193] Shiller, Z., Gwo, Y.-R., et others (1991). **Dynamic motion planning of autonomous vehicles**. *IEEE Transactions on Robotics and Automation*, 7(2):241–249.

[194] Sierra Gonzalez, D. (2019). **Towards Human-Like Prediction and Decision-Making for Automated Vehicles in Highway Scenarios**. PhD thesis, Grenoble Alpes.

[195] Sikander, G., et Anwar, S. (2018). **Driver fatigue detection systems: A review**. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2339–2352.

[196] Siouris, G. M. (2004). **Missile guidance and control systems**. Springer Science & Business Media.

[197] Stursberg, O., Fehnker, A., Han, Z., et Krogh, B. H. (2004). **Verification of a cruise control system using counterexample-guided search**. *Control Engineering Practice*, 12(10):1269–1278.

[198] Swaroop, D., et Yoon, S. M. (1999). **Integrated lateral and longitudinal vehicle control for an emergency lane change manoeuvre design**. *International Journal of Vehicle Design*, 21(2-3):161–174.

[199] Tai, L., et Liu, M. (2016). **Deep-learning in mobile robotics-from perception to control systems: A survey on why and why not**. *arXiv preprint arXiv:1612.07139*.

[200] Thornton, S. M., Pan, S., Erlien, S. M., et Gerdes, J. C. (2016). **Incorporating ethical considerations into automated vehicle control**. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1429–1439.

[201] Thrun, S. (2002). **Probabilistic robotics**. *Communications of the ACM*, 45(3):52–57.

[202] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et others (2006). **Stanley: The robot that won the darpa grand challenge**. *Journal of field Robotics*, 23(9):661–692.

[203] Toromanoff, M., Wirbel, E., Wilhelm, F., Vejarano, C., Perrotton, X., et Moutarde, F. (2018). **End to end vehicle lateral control using a single fisheye camera**. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3613–3619.

[204] Toromanoff, M., Wirbel, E., Wilhelm, F., Vejarano, C., Perrotton, X., et Moutarde, F. (2018). **End to end vehicle lateral control using a single fisheye camera**. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3613–3619.

[205] Tsugawa, S. (1994). **Vision-based vehicles in japan: Machine vision systems and driving control systems**. *IEEE Transactions on industrial electronics*, 41(4):398–405.

[206] Tsugawa, S., Yatabe, T., Hirose, T., et Matsumoto, S. (1979). **An automobile with artificial intelligence**. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2*, pages 893–895.

[207] Ulbrich, S., et Maurer, M. (2013). **Probabilistic online POMDP decision making for lane changes in fully automated driving**. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2063–2067.

[208] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., et others (2008a). **Autonomous driving in urban environments: Boss and the urban challenge**. *Journal of Field Robotics*, 25(8):425–466.

[209] Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzalez, J. P., Gowdy, J., Gutierrez, A., Harbaugh, S., Johnson-Roberson, M., Kato, H., et others (2004). **High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004**. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-37*.

[210] Urmson, C., Bagnell, J. A., Baker, C., Hebert, M., Kelly, A., Rajkumar, R., Rybski, P. E., Scherer, S., Simmons, R., Singh, S., et others (2007). **Tartan racing: A multi-modal approach to the darpa urban challenge**.

[211] Urmson, C., et others (2008b). **Self-driving cars and the urban challenge**. *IEEE Intelligent Systems*, 23(2):66–68.

[212] Valtorta, M., Kim, Y.-G., et Vomlel, J. (2002). **Soft evidential update for probabilistic multiagent systems**. *International Journal of Approximate Reasoning*, 29(1):71–106.

[213] Varshney, K. R. (2016). **Engineering safety in machine learning**. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–5.

[214] Vilca, J., Adouane, L., et Mezouar, Y. (2015). **A novel safe and flexible control strategy based on target reaching for the navigation of urban vehicles**. *Robotics and Autonomous Systems*, 70:215–226.

[215] Vilca, J., Adouane, L., Mezouar, Y., et Lébraly, P. (2013a). **An overall control strategy based on target reaching for the navigation of an urban electric vehicle**. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 728–734.

[216] Vilca, J.-M., Adouane, L., et Mezouar, Y. (2013b). **Reactive navigation of mobile robot using elliptic trajectories andeffective on-line obstacle detection**. *In Gyroscopy and Navigation Journal, Ed. Springer, Russia ISSN 2075 1087*, 4(1):14 – 25.

[217] Vilca, J.-M., Adouane, L., et Mezouar, Y. (2016). **Optimal multi-criteria waypoint selection for autonomous vehicle navigation in structured environment**. *Journal of Intelligent and Robotic Systems (JIRS)*, 82(2):301–324.

[218] Vilca, J.-M., Adouane, L., et Mezouar, Y. (2019). **Stable and flexible multi-vehicle navigation based on dynamic inter-target distance matrix**. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 20(4):1416–1431.

[219] Vilca Ventura, J. M. (2015). **Safe and flexible hybrid control architecture for the navigation in formation of a group of vehicles**. PhD thesis, Université Blaise Pascal.

[220] Wang, C., Sun, Q., Fu, R., Li, Z., et Zhang, Q. (2018a). **Lane change warning threshold based on driver perception characteristics**. *Accident Analysis & Prevention*, 117:164 – 174.

[221] Wang, P., Chan, C.-Y., et de La Fortelle, A. (2018b). **A reinforcement learning based approach for automated lane change maneuvers**. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384.

[222] Ward, E., Evestedt, N., Axehill, D., et Folkesson, J. (2017). **Probabilistic model for interaction aware planning in merge scenarios**. *IEEE Transactions on Intelligent Vehicles*, 2(2):133–146.

[223] Ward, J. R., Agamennoni, G., Worrall, S., Bender, A., et Nebot, E. (2015). **Extending time to collision for probabilistic reasoning in general traffic scenarios**. *Transportation Research Part C: Emerging Technologies*, 51:66–82.

[224] Wierwille, W. W., Hanowski, R. J., Hankey, J. M., Kieliszewski, C., Lee, S. E., Medina, A., Dingus, T. A., et others (2002). **Identification of driver errors: overview and recommendations.** Technical Report, United States. Federal Highway Administration.

[225] Wikipedia (2020). **CMA-ES**, Consulted on July 2020: https://en.wikipedia.org/wiki/CMA-ES.

[226] Woo, H., Ji, Y., Kono, H., Tamura, Y., Kuroda, Y., Sugano, T., Yamamoto, Y., Yamashita, A., et Asama, H. (2017). **Lane-change detection based on vehicle-trajectory prediction**. *IEEE Robotics and Automation Letters*, 2(2):1109–1116.

[227] Worrall, R., et Bullen, A. (1970). **An empirical analysis of lane changing on multilane highways**. *Highway Research Record*, (303).

[228] Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., et Sumner, A. (2000). **Torcs, the open racing car simulator**. *Software available at http://torcs. sourceforge. net*, 4(6):2.

[229] Xu, B., et Li, Q. (2016). **A spatial logic for modeling and verification of collision-free control of vehicles**. In *Engineering of Complex Computer Systems (ICECCS), 2016 21st International Conference on*, pages 33–42.

[230] Xu, H., Gao, Y., Yu, F., et Darrell, T. (2017). **End-to-end learning of driving models from large-scale video datasets**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182.

[231] You, C., Lu, J., Filev, D., et Tsiotras, P. (2018). **Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning**. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1227–1232.

[232] Young, K. L., et Salmon, P. M. (2012). **Examining the relationship between driver distraction and driving errors: A discussion of theory, studies and methods**. *Safety science*, 50(2):165–174.

[233] Yuan, Q., Gao, Y., et Li, Y. (2016). **Suppose future traffic accidents based on development of self-driving vehicles**. In *International Conference on Man-Machine-Environment System Engineering*, pages 253–261. Springer.

[234] Zhang, J., et Roessler, B. (2009). **Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving**. In *Autonome Mobile System 2009*, pages 249–258. Springer.

[235] Ziegler, J., Bender, P., Dang, T., et Stiller, C. (2014a). **Trajectory planning for bertha—a local, continuous method**. In *2014 IEEE intelligent vehicles symposium proceedings*, pages 450–457.

[236] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., et others (2014b). **Making bertha drive—an autonomous journey on a historic route**. *IEEE Intelligent transportation systems magazine*, 6(2):8–20.

[237] Ziegler, J., Lategahn, H., Schreiber, M., Keller, C. G., Knöppel, C., Hipp, J., Haueis, M., et Stiller, C. (2014c). **Video based localization for bertha**. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1231–1238.

[238] Zyner, A., Worrall, S., et Nebot, E. (2018). **A recurrent neural network solution for predicting driver intention at unsignalized intersections**. *IEEE Robotics and Automation Letters*, 3(3):1759–1764.

Abstract:

Recent advances in Autonomous Vehicles (AV) driving raised up all the importance to ensure the complete reliability of AV maneuvers even in highly dynamic and uncertain environments/situations. This objective becomes even more challenging due to the uniqueness of every traffic situation/condition. To cope with all these very constrained and complex configurations, AVs must have appropriate control architecture with reliable and real-time Risk Assessment and Management Strategies (RAMS). These targeted RAMS must lead to reduce drastically the navigation risks (theoretically, lower than any human-like driving behavior), with a systemic way. Consequently, the aim is also to reduce the need for too extensive testing (which could take several months and years for each produced RAMS without at the end having absolute prove). Hence the goal in this Ph.D. thesis is to have a provable methodology for AV RAMS. This dissertation addresses the full pipeline from risk assessment, path planning to decision-making and control of autonomous vehicles. In the first place, an overall Probabilistic Multi-Controller Architecture (P-MCA) is designed for safe autonomous driving under uncertainties. The P-MCA is composed of several interconnected modules that are responsible for: assessing the collision risk with all observed vehicles while considering their trajectories' predictions; planning the different driving maneuvers; making the decision on the most suitable actions to achieve; control the vehicle movement; aborting safely the engaged maneuver if necessary (due for instance to a sudden change in the environment); and as last resort planning evasive actions if there is no other choice. The proposed risk assessment is based on a dual-safety stage strategy. The first stage analyzes the actual driving situation and predicts potential collisions. This is performed while taking into consideration several dynamic constraints and traffic conditions that are known at the time of planning. The second stage is applied in real-time, during the maneuver achievement, where a safety verification mechanism is activated to quantify the risks and the criticality of the driving situation beyond the remaining time to achieve the maneuver. The decision-making strategy is based on a Sequential Decision Networks for Maneuver Selection and Verification (SDN-MSV) and corresponds to an important module of the P-MCA. This module is designed to manage several road maneuvers under uncertainties. It utilizes the defined safety stages assessment to propose discrete actions that allow to: derive appropriate maneuvers in a given traffic situation and provide a safety retrospection that updates in real-time the ego-vehicle movements according to the environment dynamic, in order to face any sudden hazardous and risky situation. In the latter case, it is proposed to compute the corresponding low-level control based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) that allows the ego-vehicle to pursue the advised collision-free evasive trajectory to avert an accident and to guarantee safety at any time. The reliability and the flexibility of the overall proposed P-MCA and its elementary components have been intensively validated, first in simulated traffic conditions, with various driving scenarios, and secondly, in real-time with the autonomous vehicles available at Institut Pascal.

Keywords: Autonomous driving, Multi-controller architectures, Risk assessment and management, Decision-making under uncertainty, Motion-planning, Safety verification.

Résumé :

Les dernières avancées en matière de conduite de véhicules autonomes (VAs) ont fait apparaître toute l'importance de garantir la fiabilité complète des manœuvres que doivent effectuer les VAs, y compris dans des environnements/situations très dynamiques et incertains. Cet objectif devient encore plus ardu en raison du caractère unique de chaque situation/condition de circulation. Pour faire face à toutes ces configurations très contraignantes et complexes, les VAs doivent disposer d'une architecture de contrôle appropriée avec des Stratégies d'Evaluation et de Gestion des Risques (SEGR) fonctionnant en temps-réel et d'une manière fiable. Ces SEGR ciblées doivent conduire à une réduction drastique des risques de conduite. Théoriquement et de maniéré systémique, ces SEGR doivent aboutir à un risque de conduite inférieur à tout comportement de conduite humaine. En conséquent, il est également question de réduire la nécessité d'effectuer des tests très poussés, qui peuvent prendre plusieurs mois/années pour au final ne pas avoir de preuves formelles de la viabilité et de la sûreté complète du système. Ainsi, les travaux présentés dans cette thèse de doctorat ont pour but d'avoir une méthodologie prouvable pour les SGER des VAs. Cette thèse porte sur l'ensemble du processus, en partant de l'évaluation des risques, de la planification de la trajectoire jusqu'à la prise de décision et au contrôle du véhicule autonome. En premier lieu, une architecture multi-contrôleurs probabiliste (Probabilistic Multi-Controller Architecture P-MCA) est conçue pour une conduite autonome sûre en présence d'incertitudes. Cette architecture est composé de plusieurs modules interconnectés qui sont responsables de : l'évaluation du risque de collision avec tous les véhicules observés tout en considérant les prévisions de leurs trajectoires ; la planification des différentes manœuvres de conduite ; la prise de décision sur les actions les plus appropriées à réaliser ; le contrôle du mouvement du véhicule ; l'interruption en toute sécurité de la manœuvre engagée si nécessaire (en raison par exemple d'un changement soudain de l'environnement routier) ; et en dernier recours la planification des actions évasives à défaut d'un autre choix. L'évaluation des risques proposée est basée sur une stratégie à deux étapes. La première étape consiste à analyser la situation actuelle de conduite et à prévoir les éventuelles collisions. Cette étape est réalisée en tenant compte de plusieurs contraintes dynamiques et des conditions de circulation connues au moment de la planification. La deuxième étape est appliquée en temps-réel, durant la réalisation de la manœuvre, où un mécanisme de vérification de la sécurité est activé pour quantifier les risques et la criticité de la situation de conduite sur le temps restant pour réaliser la manœuvre. La stratégie décisionnelle est basée sur un réseau Bayésien de décision à niveaux séquentiels pour la sélection et la vérification des manœuvres (Sequential Decision Networks for Maneuver Selection and Verification SDN-MSV) et constitue un module essentiel de l'architecture P-MCA. Ce module est conçu pour gérer plusieurs manœuvres routières dans un environnement incertain. Il utilise l'évaluation des étapes de sécurité définies pour proposer des actions discrètes qui permettent de : réaliser des manœuvres appropriées dans une situation de trafic donnée, il fournit également une rétrospective de la sécurité, cette dernière actualise en temps-réel les mouvements de l'égo-véhicule en fonction de la dynamique de l'environnement, afin de faire face à toute situation dangereuse et risquée soudaine. Dans ce dernier cas, il est proposé de calculer le contrôle de bas niveau correspondant basé sur une stratégie d'évolution avec adaptation de matrice de covariance (Covariance Matrix Adaptation Evolution Strategies CMA-ES) qui permet à l'égo-véhicule de suivre la trajectoire d'évitement sans collision conseillée pour éviter un accident et garantir la sécurité à tout moment. La fiabilité et la flexibilité de l'ensemble de l'architecture P-MCA proposée et de ses composants élémentaires a été validé de manière intensive, d'une part dans des conditions de circulations simulées, avec différents scénarios de conduite, et d'autre part, en temps-réel avec les véhicules autonomes disponibles à l'Institut Pascal.

Mots-clés : Conduite autonome, Architecture multi-contrôleurs, Prise de décision dans l'incertain, Planification de trajectoire, Évaluation des risques, Vérification de la sécurité.