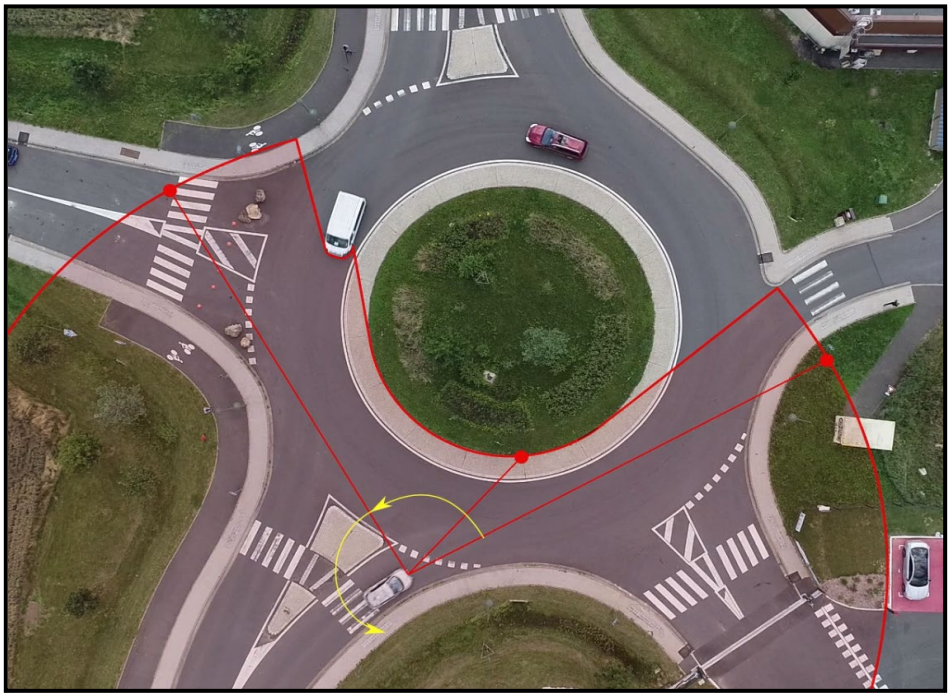
Safe autonomous vehicles navigation in roundabouts with cooperative perception from an intelligent infrastructure

Stefano Masi

Par **Stefano MASI**

Safe autonomous vehicles navigation in roundabouts with cooperative perception from an intelligent infrastructure

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 5 mai 2021
Spécialité : Sciences et Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253)
D2606

Safe Autonomous Vehicles Navigation in Roundabouts with Cooperative Perception from an Intelligent Infrastructure

Stefano Masi

PhD thesis prepared in the Heudiasyc Laboratory, UMR UTC-CNRS 7253 and presented for the degree of doctor of the University of Technology of Compiègne.

Spécialité : Sciences et Technologies de l'Information et des Systèmes

Defended on on May, 5th 2021:

Thesis committee:

Reviewers:	Prof. Eduardo Nebot	University of Sydney
	Prof. Huijing Zhao	Peking University
Examiners:	Prof. Roberto Sacile	Università degli Studi di Genova
	Dr. Sio-Song Ieng	Université Gustave Eiffel
	Dr. Javier Ibanez-Guzman	Groupe Renault
President	Prof. Véronique Cherfaoui	Université de Technologie de Compiègne
Thesis advisor:	Prof. Philippe Bonnifait	Université de Technologie de Compiègne
Thesis co-advisor:	Dr. Philippe Xu	Université de Technologie de Compiègne

heudiasyc

ALLIANCE SORBONNE UNIVERSITÉ

utc Recherche

cnrs dépasser les frontières

Contents

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Philippe Bonnifait and my co-advisor Dr. Philippe Xu for the continuous support of my Ph.D study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentor for my Ph.D study.

Besides my advisors, I would like to warmly thank my thesis committee: Prof. Huijing Zhao and Prof. Eduardo Nebot for their careful review of the manuscript, Prof. Véronique Cherfaoui, Prof. Roberto Sacile, Dr. Javier Ibanez-Guzman and Dr. Sio-Song Ieng for taking the time to read and review my research work. Their insightful comments gave me the opportunity to improve my research and provided me new ideas and pointers for new future perspectives.

I am deeply grateful to the researchers, professors, engineers and the other Ph.D. students of the Heudiasyc and SivaLab laboratories. Again, their support has been precious in countless occasions for the accomplishment of both the methods and the experiments that I presented in this manuscript. Moreover, I am really grateful for their support for helping me in preparing the numerous demonstration sessions that have been organized during my Ph.D. Such demonstrations provided both excellent results and helpful feedback to validate and improve my research work.

A special thanks also goes to my former internship student Dr. Edoardo Bernardi, who decided to do its internship with Dr. Philippe Xu and I. The quality of the work and the results achieved in this internship have been really useful for several parts of the research works presented in this manuscript.

My sincere thanks also goes to all the members of the Tornado project for offering me the opportunity to integrate my researches in this project and for providing me several occasions to test my research prototypes in a real-life environment. This has been extremely useful to validate the main methods and to add more value to the experimental results of this work.

Last but not the least, any attempt at any level couldn't be satisfactorily completed without the support and guidance of my family, my girlfriend and my friends. For this reason, I would like to express my special thanks of gratitude to my parents, my brother, my grandmother, my girlfriend and all my dearest friends, who always encourage me and who provided me a solid support during these years.

Abstract

Although autonomous vehicle technology has evolved significantly in recent years, the navigation of self-driving vehicles in complex scenarios is still an open issue. One of the major challenges in these conditions is safe navigation on roads open to public traffic. In such driving environments, the main issue is the interaction of the autonomous vehicle with regular traffic, as the behaviors and intentions of human-driven vehicles are hard to predict and understand.

In this PhD thesis, we propose and study an approach to ensure safe autonomous driving in such scenarios. In particular, we study the contribution that an intelligent infrastructure can provide to improve safety via a reliable road users detection during the navigation into complex zones. In this work, we address this problem at different levels, considering a roundabout insertion as the main case study.

In the first part, we develop a navigation algorithm that adapts the concept of virtual platooning to roundabout crossing, relying on High Definition (HD) maps. Indeed, HD maps represent a useful framework to predict collision at lane-level in terms of colliding trajectories with the ego-vehicle navigation corridor. The computations are done in curvilinear abscissa which is particularly efficient and fast. Moreover, to cope with human-driven vehicles unknown intentions, virtual instances of the vehicles combined by an interval-based occupancy representation are proposed. These intervals also include localization and perception uncertainties. The method safely avoids collisions and guarantees that no priority constraints are violated during the insertion maneuver, without providing an overly cautious insertion policy. Finally, we also propose some techniques to extend the navigation strategy to the safe crossing of multi-lane roundabouts.

The performance of this strategy is evaluated using the SUMO simulation framework. To evaluate the complexity of the simulation scenario, a highly interactive vehicle flow is generated using real dynamic traffic data from a public dataset. We also report real tests carried out with an experimental self-driving vehicle on a test circuit with both a simulated and real traffic.

In the second part, we focus on consistency and robustness of road users detection. The main objective is to implement a LiDAR-based road users detection pipeline to sense the driving environment and provide the necessary information to perform autonomous driving. To do so, a fast and consistent method to manage uncertainties on detected traffic agents is presented. The information provided by a 3D LiDAR-based object detector is combined with an HD map to identify the drivable space of the carriageway. A novel approach propagating localization uncertainty in the LiDAR points is proposed and the performance of this approach is evaluated both in an Euclidean and in a map-based curvilinear reference frames thanks to the use of real data acquired at the entrance of a roundabout.

In the third part, we improve the performance of the on-board perception system by using a map-aided tracking algorithm which provides the speed of road users and an better estimation of their occupancy at lane level. To enhance the estimation process, we study how the information provided by a remote cooperative intelligent infrastructure can help the perception of a self-driving vehicle. To do so, a

multi-sensor data fusion approach integrates the infrastructure information in the perception framework. This extension leads to two main advantages. First, it enlarges the field of view of the on-board perception system of the vehicle. Second, the combined system perception provides a more accurate state estimation of perceived objects in the zones where the fields of view overlap. It brings also redundancy which increases integrity. The improvement of the cooperative system compared to the standalone one is evaluated in terms of visibility but also in terms of state and occupancy estimation.

The contributions of the PhD thesis can help intelligent vehicles to improve their autonomous navigation capabilities in complex situations where reduced visibility and interactions with other road users rise safety critical issues.

Résumé

Bien que la technologie des véhicules autonomes ait considérablement évolué au cours des dernières années, la navigation des véhicules autonomes dans des scénarios complexes reste toujours une question ouverte. L'un des principaux défis dans ces scénarios est la sécurité de la navigation sur les routes ouvertes à la circulation publique. Dans ces environnements de conduite, le principal problème est l'interaction du véhicule autonome avec les véhicules à conduite manuelle, car leurs comportements et leurs intentions sont difficiles à prévoir et à comprendre.

Dans cette thèse, une approche est proposée pour assurer une conduite autonome sécurisée dans ces scénarios. En particulier, nous avons étudié la contribution qu'une infrastructure intelligente digne de confiance peut apporter pour améliorer la sécurité grâce à une détection fiable des usagers de la route lors de la navigation dans des zones complexes. Dans ce travail, nous abordons ce problème à différents niveaux, en considérant la manœuvre d'insertion dans un rond-point comme cas d'étude principal.

Dans la première partie, un algorithme de navigation est développé selon le concept de suivi virtuel adapté aux ronds-points, en s'appuyant sur des cartes haute définition (HD). En effet, les cartes HD représentent un bon outil pour prédire les interactions au niveau de la voie en termes de trajectoires de collision avec le couloir de navigation de l'ego-véhicule. Les calculs se font en abscisses curvilignes ce qui est particulièrement efficace et rapide. Pour faire face aux intentions inconnues des véhicules à conduite manuelle, des instances virtuelles de véhicules dont l'occupation est représentée avec des intervalles sont proposées. Ces intervalles englobent également les incertitudes de localisation et de perception. La méthode apporte ainsi un haut niveau de sécurité vis-à-vis des collisions et garantit qu'aucune contrainte prioritaire n'est violée lors de la manœuvre d'insertion, et ce sans fournir une politique d'insertion trop prudente. Enfin, nous proposons également des techniques pour étendre cette stratégie de navigation à la traversée sure de ronds-points à plusieurs voies. La performance de cette stratégie a été évaluée à l'aide de l'outil de simulation SUMO. Un flux de véhicules hautement interactif a été généré à l'aide de données de trafic dynamiques réelles provenant d'un jeu de données publiques. Nous avons également effectué des tests réels avec un véhicule expérimental à conduite autonome sur un circuit de test avec un trafic simulé et réel.

Dans la deuxième partie de la thèse, la cohérence et la robustesse de la détection des usagers de la route avec un capteur LiDAR omnidirectionnel et multi-faisceaux sont étudiées. Un pipeline complet de détection des usagers de la route basé sur le LiDAR est proposé afin de détecter l'environnement dynamique et afin de fournir les informations nécessaires à la conduite autonome. Les informations fournies par un détecteur d'objets 3D basé sur le LiDAR sont combinées avec une carte HD pour identifier l'espace occupé par les usagers sur la route. Une nouvelle approche pour propager l'incertitude de localisation dans les points LiDAR est proposée et la performance de cette approche est évaluée expérimentalement en termes d'occupation en considérant deux représentations, l'une euclidienne et l'autre en coordonnées curvilignes.

Dans la troisième partie, les performances du système de perception embarqué sont améliorées avec un algorithme de pistage assisté par la carte, ce qui est important pour estimer la vitesse des usagers de la route avec un capteur optique. Pour améliorer le processus d'estimation, nous étudions comment les informations fournies par une infrastructure intelligente coopérative à distance peuvent aider la perception embarquée d'un véhicule autonome. Pour cela, une méthode de fusion de données multi-capteurs capable de gérer les latences des données transmises par l'infrastructure est proposée. Cette extension conduit à deux avantages principaux : elle élargit le champ de vision du système de perception embarqué du véhicule et elle apporte de la redondance ce qui augmente de l'intégrité. L'amélioration du système coopératif par rapport au système autonome est évaluée en termes de visibilité mais aussi en termes d'estimation d'état et d'occupation.

Les contributions de cette thèse de doctorat peuvent aider les véhicules intelligents à améliorer leurs capacités de navigation autonome dans des situations complexes où la visibilité réduite et les interactions avec les autres usagers de la route posent des problèmes notamment en termes de sécurité.

1 General Introduction

Autonomous driving is a widely studied research topic and, in the recent years, several progresses have been made on this subject particularly in urban environments. Nowadays there are several examples of vehicles that are able to perform navigation tasks in full autonomous mode on different kinds of driving environments.

To perform autonomous driving in urban traffic, several aspects are still open issues. One of the most challenging ones is the lack of visibility in complex urban scenarios. This means that, for the autonomous vehicle, it is not always possible to obtain a complete and reliable knowledge of the other traffic participants. This lack of information might lead to poor performance when the autonomous vehicles navigates through a complex urban environment. In particular, the presence of missing information about other road users might compromise the safety during navigation. In other words, the risk of accidents arises if the autonomous vehicle has a partial knowledge of the driving environment.

To overcome this issue, several approaches can be found in the literature. One appealing solution is offered by vehicle-to-vehicle (V2V) communication, which allows vehicles to broadcast information about their state to other traffic participants. This has the advantage to provide mutual awareness between vehicles, covering occluded zones and blind spots in the autonomous vehicle perception system. However, nowadays this technology is available only on few vehicles.

Another solution is provided by the leveraging of external information sent from a remote intelligent infrastructure. Such a system has the advantage of providing an extra source of information from a different perspective with respect to the on-board perception system of the autonomous vehicle. This information can be combined together with the one provided by the on-board perception system to obtain a more complete and reliable representation of the driving environment.

Another aspect that is crucial for autonomous vehicles navigation in urban traffic is to deal with unknown vehicles intentions. This situation is common when the autonomous vehicle navigates in an environment where several manually driven vehicles are present. In this case, under the hypothesis that no direct communication between vehicles exists, it is often challenging to estimate and understand other traffic participants intentions. The only information that the self-driving vehicle can have from other vehicles is the one obtained by its own perception system. This implies that, in many cases, the estimation of some crucial pieces of information as the other vehicles trajectories and intentions is difficult to estimate. As a consequence, navigation algorithms must be designed taking into account this constraint, in a way that they should be able to operate with incomplete or missing information. To do so, it is required to find a way to incorporate not only uncertainty about vehicles intentions, but also uncertainty about vehicles spatial occupancy directly into the navigation strategy.

Once this step is achieved, it is necessary to provide a consistent way to validate the navigation strategy. Starting from the fact that real tests with other road users offer the most realistic way of validation, they are not always possible in public roads. Real driving tests performed in private test circuit give an overview of the

behavior of the algorithm on a real system. However, they are often limited to use-cases with a limited number of interaction with other road users and they suffer of scalability issues.

To get over this issue, it is necessary to re-create a realistic behavior in a simulated environment. This is necessary because nowadays driving simulators always present a gap between simulated and real driving environments in particular with several road users in interaction. In addition, to provide a solid test-bed for navigation algorithms, it is necessary to re-create a realistic driving behavior starting from real traffic data.

Finally, it is necessary to find a criterion to quantify the level of uncertainty about the estimated occupancy of perceived objects. This criterion has to provide uncertainty bounds according to a given risk. Integrity metrics of localization are quite well known nowadays and we are looking for a method to extend this concept to perception. However, this task is not simple because perception is a composite task that relies on several sub-tasks. All pre-treatments must therefore be given special attention.

1.1 Problem Statement

1.1.1 Main Objectives

The main problem addressed in this PhD thesis is the safe navigation of autonomous vehicles in complex urban scenarios with several road users in interaction. The main scenario that is considered throughout this work is the autonomous vehicle (AD) navigation in roundabouts. For this driving situation, a reliable estimation of the occupancy and speed of the perceived objects obtained by the perception system is a key issue. As a case study, we consider the roundabout crossing maneuver, where the self-driving car has to cross safely the roundabout with an arbitrary flow of regular vehicles inside it.

To cope with this, a navigation algorithm has been designed to perform safe roundabouts navigation. It uses a High Definition (HD) map and a map-based curvilinear framework. The proposed navigation strategy is entirely based on this curvilinear framework and on the concept of virtual platooning. This method, that was initially designed to cope with autonomous vehicles cooperative intersection crossing, has been extended to roundabout crossing with priority constraints.

Another crucial point is the occupancy estimation of the perceived road users. As a matter of fact, to avoid collisions due to wrongly estimated localization of the other traffic participants, we propose to extend the curvilinear formalism with an interval-based formalism to represent uncertainties on vehicles occupancy and we present a way to provide a good estimation of the portion of the road occupied by each perceived object. Moreover, we develop an integrity criterion to quantify the overall level of uncertainty of the perceived objects according to a given risk.

Collaboration with an intelligent infrastructure is a new way to enhance safety during navigation. We propose to investigate a cooperative data fusion of the on-board perception with information provided by a remote intelligent infrastructure that enhance both the self-driving car knowledge of the driving environment and the integrity estimation of the perceived obstacles.

In the following part of this chapter, we consider in more detail the main problems addressed in this manuscript.

1.1.2 Autonomous Vehicles Navigation in Roundabouts

To navigate in urban traffic, an autonomous vehicle has to complete a series of navigation missions (e.g. going from a point A to a point B without collision) while a dynamic flow of vehicles is present on the driving scene. This kind of scenario is typical for robotaxis and mobility oriented autonomous driving applications. In such a situation, we consider the other traffic participants as an adversarial traffic flow. Traffic is said adversarial if, under some circumstances, the other road users do not cooperate with the autonomous vehicle. On the contrary, there is an individual and selfish behavior regarding the adopted driving strategy. Such constraint also implies that there is here no V2V or intention sharing between cars. In this case, the challenge is twofold: first, autonomous vehicle has to both detect the other vehicles and then, it has to estimate their intentions only by means of its on-board perception system.

One of the main difficulties that arise is the necessity to quantify the level of uncertainty of perceived environment and how to take into account such uncertainty directly into the navigation framework. It is our opinion that, to enhance the safety of AD vehicle navigation, the navigation strategy must be able to encapsulate in the decision-making process the uncertainty associated to traffic participants. Such uncertainty can be categorized in the following two groups:

1. Uncertainty about vehicles occupancy and speeds estimation;
2. Uncertainty about the identification of drivers intentions.

In the first case, uncertainty is linked to the accuracy of the estimation of the road users localization and velocity, while in the second case, uncertainty refers to the prediction or the identification of a certain driving maneuvers in a driving environment.

It is widely known that a good knowledge of these information with low uncertainty leads to a better driving performance in terms of safety. Furthermore, predicting the intentions of an adversarial vehicle can lead to more optimized navigation maneuvers and a more fluid navigation among other cars.

To summarize, a safe autonomous vehicles navigation method must meet the following requirements:

1. Ensure safe navigation maneuvers in a complex scenario;
2. Provide robustness to uncertain road users detections;
3. Provide a smooth and not overly-cautious navigation strategy.

Another problem is the test and validation of navigation strategies in a realistic simulated environment. Nowadays there exist a wide range of tools and frameworks to provide simulation for autonomous driving. Dedicated simulators provide a simulated representation for different purposes (perception, navigation, routing, etc.) with different levels of detail. The use of simulation to test the performance of navigation algorithms is a widely used technique in the AD vehicles domain for several reasons. On one hand, it provides a cheap and scalable way to validate developed strategies, allowing the generation of a wide range of different scenarios under different conditions. On the other hand, it ensures safety during tests and avoids the occurrence of accident and dangerous driving situations, preventing from the risk of causing damages to systems and harming people.

However, even if the level of precision and detail has been increased over the years, a gap between simulated environments and real tests still exists. In general, such a gap is mostly due to the poor quality of simulated uncertainty on data and the pseudo-randomness of simulated uncertainty. Regarding traffic flows simulators, this gap is mostly due to the simplified modeling of adversarial vehicles behaviors and limited motion models. Indeed, it is hard to describe with a mathematical model the wide set of possible human behaviors during navigation. Some safety critical maneuvers as for example lane change or overtaking are still far from the naturalistic way of driving. For this reason, in the recent years, some researchers prefer to validate their algorithms on recorded datasets rather than on simulated scenarios. This, in general, provides a more realistic performance of algorithms because the data are closer (sometimes the same) to real ones, especially for complex navigation maneuvers and interaction between road agents. However, the main drawback is the lack of reactivity and interaction feedback to the autonomous vehicle navigation strategy. Indeed, a recorded flow cannot produce any reaction w.r.t. the autonomous vehicle because such a vehicle was not present at the moment of the dataset recording. This lack of reaction does not permit to take into account the behavior of other vehicles w.r.t. the autonomous vehicle chosen decision.

In this work, we will combine the two approaches. Simulations will be carried out for decision and navigation algorithms while raw data records will be used for perception systems.

1.1.3 Road Users Detection and Localization

One other aspect that is crucial for autonomous navigation is the robust and reliable detection of road users. If there is no direct communication between the self-driving vehicle and the other vehicles, the navigation method must rely only on its perception capabilities for environment sensing and road obstacles detection tasks.

For this reason, another problem addressed in this manuscript is the design and development of a perception system that can provide the navigation layer the necessary information about the surrounding driving environment

Such a system must be able to operate in a real-time context and to provide detection results sufficiently fast to react to any dynamical behavior or sudden change in the driving scene. Moreover, as we consider safety-critical navigation tasks, it is also required to avoid as much as possible missed detections, which may represent an issue from the safety point of view.

In order to improve both the detection performance and the real-time feasibility, High Definition (HD) maps can be exploited by the perception system. One of the main advantages of this technology is their capability of identifying the drivable surface. This is useful to filter detected objects with respect to the drivable surface. Using a map implies being able to localize precisely the AD vehicle inside the map itself. This could lead to dangerous results if the localization is not accurate. Therefore the localization uncertainty needs to be carefully taken into account in order not to wrongly filter the detected objects.

This manuscript presents the design and development of a perception system that provides the navigation layer the necessary information about the surrounding driving environment using 3D LiDAR and a HD map. Integrity of the information will be considered carefully.

1.1.4 Integrity of Perceived Information for Autonomous Driving

The concept of integrity has been proposed first to provide a means to bound localization error and sensor faults for flight localization. This technique consists in providing a real-time measure of the level of trust to be placed in the localization estimates as vehicles operate [4,95]. In other words, it provides a means for knowing whether the position estimates computed by the AD vehicle on-board system are exploitable for navigation purposes, based on a given risk factor. This concept has been applied to the AD vehicles domain to quantify the accuracy of GNSS-based absolute positioning systems [4].

Based on the knowledge of some uncertainty metrics, such as a covariance matrix for low requirement levels, it is possible to compute some confidence bounds on the position uncertainty according to a given risk level α. The integrity requirement implies that the true position should be within the bounds with a probability of at least $1-\alpha$. The intervals bounds obtained in such way represent the probability that the ground truth of the vehicle position is contained in such interval with probability of $1-\alpha$.

Bearing in mind this concept, one of the main objectives of this work is to discuss and propose a method that allows to extend and exploit the same reasoning to quantify the level of confidence on the AD vehicle perception results. However, the application of the aforementioned method to perception is not straightforward due to the complexity and the layered structure of the perception task.

A first definition for this concept can be to define the integrity of perceived information as a measure of the risk caused by an unperceived object. This definition allows to quantify the risk that the missed detection of an obstacle can have on a certain driving task, according to the complexity of the driving scenario. However, such a definition is not easy to implement in practical applications. This comes from the fact that, in general, environment perception is a composite task and such a definition must be applied to all the different sub-layers [18]. Furthermore, in order to apply such definition to different sub-layers of the perception, one needs to individuate and identify these sub-layers.

According to [18], perception integrity can be seen as a measure of the certainty of the knowledge represented by the perception system in a given area. This definition is useful to quantify how the available information from the different sensors is disposed in the autonomous vehicle surrounding environment. Furthermore, it also includes the presence of zones where information is unavailable or unknown. As we can see, this definition covers only some aspects of the ideal definition that we proposed before. To overcome this, we propose to define in a more concrete and operational way perception integrity at different levels:

1. Integrity of the perception in terms of localization uncertainty of the road users;
2. Integrity of the detected free-space;
3. Integrity of the perception result in terms of classification of the objects;
4. Integrity of perceived objects occupancy on the road surface.

In case (1), the main idea is to first bound localization errors of the ego-vehicle according to a given risk, then to propagate these localization uncertainty bounds to the estimation errors of a frame attached to the perceived objects.

In case (2), the focus of integrity is on the perceived free space rather than computing the integrity of detected objects. Case (2) aims to quantify the zones of the driving environment where we are sure that only the ground has been detected and nothing more.

Case (3) corresponds to the risk caused by an object wrongly classified or to quantify the probability that an object has been correctly classified. In fact, some detection algorithms for images are able to provide information not only on the classification of an object, but also a confidence score.

Finally, in case (4), we focus the attention on determining objects spatial occupancy according to a given risk which is well adapted to vehicles whose size is to be taken into account (a bus for example). It can be thought that case (4) is the dual of the situation presented in case (2). However, this is not the case. The main reason is that there are blind spots and hidden areas. In this research work, we will be particularly interested in case (4) because it fits perfectly with the navigation algorithm we use. In addition, we will limit ourselves to 1D occupancy along the polylines of the map.

Moreover, when a HD map supports the navigation strategy, the integrity of the information contained in the map itself is an important question. However, this topic is out of the scope of this dissertation and we suppose that the the HD map does not contain misleading or wrong information.

1.1.5 Multi-sensor and Cooperative Data Fusion

Another aspect that is treated in this manuscript is the cooperative perception with a remote intelligent infrastructure system. The main idea is to exploit a cooperative perception system to provide a reliable and consistent information to the AD vehicle. The perception system consists of a remote and fixed infrastructure that senses only some particular driving zones. In other words, neither the position, nor the field of view of the infrastructure vary over time. To do so, we focus the attention on the cooperative data fusion between these two sources of information. In a cooperative system, data fusion may be performed at different levels, with filtering and data association issues. A problem that arises in this context concerns the data fusion of asynchronous and out-of-sequence data. In a cooperative system with remote sensors involved, it is necessary to guarantee the synchronization of the clocks of all the systems to be able to solve those problems and to perform a proficient data fusion process.

An aspect that we study in this part of the work is the performance of the cooperative system in terms of integrity of estimated objects occupancy, compared with the one obtained by using only the on-board perception. In particular, we would like to quantify the infrastructure contribution to this aspect according to the following criteria:

- Gain with respect to the enhancement of the AD perception system field of view;
- Gain with respect to the accuracy of estimated objects.

In the former case, the infrastructure information contributes to enlarge the self-driving car field of view, while in the latter case, the infrastructure detection can be useful to better estimate the objects occupancy and to ensure safety during AD vehicle navigation.

We need to underline that in this work we suppose that no misleading, erroneous or malicious information is sent by the infrastructure. In other words, we rely on the infrastructure information and all the problems that revolve around the trustworthiness of the infrastructure are out of the scope of this work.

Finally, Figure 1.1 figures out the architecture of the whole system. For each module that constitutes the system pipeline, an explanation of its role is reported in the following section.

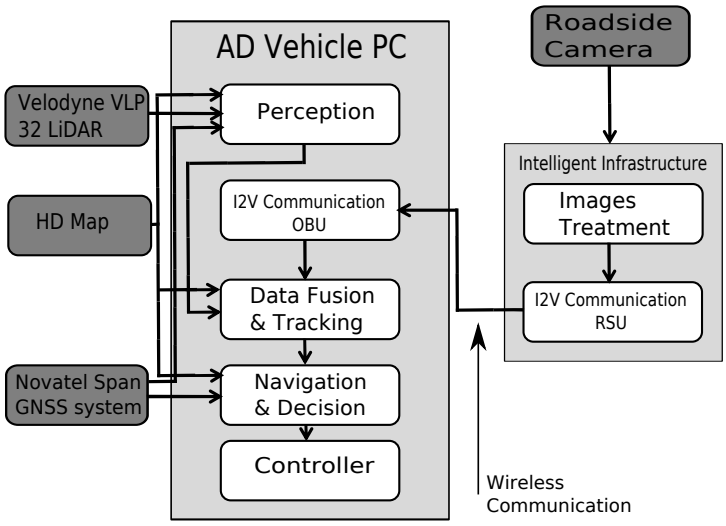


Figure 1.1: System architecture considered in this work. Notice that the part relative to the intelligent infrastructure will be used only in the last part of the work.

1.2 Scientific Contributions and Manuscript Organization

In chapter 2, a presentation of the state-of-the-art of autonomous vehicles navigation, focusing in navigation algorithms designed for roundabout navigation, local trajectory planning, high definition maps and traffic simulation is given. Furthermore, the main formalisms and concepts that will be used through the rest of this work will be discussed. In particular, the following elements will be highlighted:

- A high definition map formalism to compute curvilinear coordinates along maps polylines;
- A traffic flow simulation tools and dataset for autonomous driving.

The contents and the results presented in this chapter lead to the publication of a conference paper at the IEEE Intelligent transportation Systems conference [40].

Chapter 3 is focused on the navigation strategy for roundabout crossing. First, the completely cooperative use-case is taken into account as a proof of concept for the algorithm. Then, the previous case has been extended to the case of an autonomous vehicle that navigates among a regular vehicle traffic flow. The main contributions of this part are listed hereafter:

- A safe, priority-preserving and not overly cautious decision method for an AD vehicle crossing a two-lane roundabout among a vehicle flow;

- A practical evaluation of the algorithm in terms of safety and fluidity of the insertion, with realistic simulations and real experiments.

The contents and the results presented in this chapter lead to two conference papers at the IEEE Intelligent Vehicles Symposium [66, 67] and to the publication of a journal paper in the IEEE Transactions on Intelligent Transportation Systems [68].

In chapter 4, a geometric LiDAR-based road users detection method with a focus on how a HD map can be integrated in the perception task is described in details. For that purpose, we provide an approach that is robust w.r.t. localization uncertainty. The main contribution of this part are listed hereafter:

- A fast LiDAR-based road users detection that captures the occupancy uncertainty and the localization errors;
- A HD map-based filtering step to reject obstacles that do not belong to the road surface.

The main contributions presented in this chapter have been also published in a IEEE Intelligent Vehicles Symposium [13].

Chapter 5 is dedicated to LiDAR-based tracking and objects occupancy estimation and cooperative multi-sensor data fusion. In particular, the fusion of on-board vehicle perception with information broadcast from a remote and intelligent infrastructure is taken into account. The goal is to enhance perception performance in terms of objects occupancy estimation. The main contributions of this part are listed hereafter:

- The development of a LiDAR-based tracking method to estimate objects speed and occupancy;
- The cooperative data fusion between LiDAR and infrastructure.

The contents and the results presented in this chapter lead to the publication of a paper at the IEEE Intelligent transportation Systems conference.

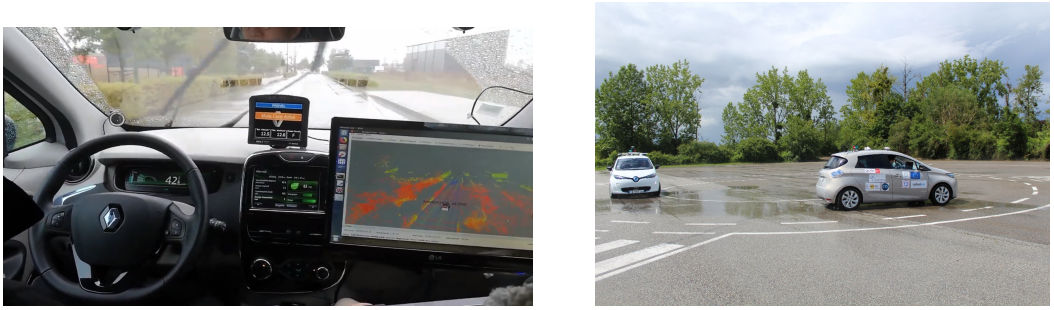
In chapter 6, conclusions about the different research topics investigated in this work are drawn, focusing the attention on the contributions of HD maps and the intelligent infrastructure to improve autonomous vehicles navigation.

Finally, Appendix 7 proposes two sections to better understand how the experimental system works. In the first appendix, we analyze the ETSI communication standards (CAM, DENM and CPM) in order to understand which standard has to be used to implement I2V communication.

In the second part, an overview on the dataset that has been used for our test is given. Such dataset contains several driving scenarios where several vehicles and an intelligent infrastructure can exchange between them their local perception in order to augment their knowledge of the driving scenario. Such dataset has been recorded to validate algorithms based on shared perception.

1.3 Other Contributions

This PhD thesis has been carried out in the context of a French national project entitled “Tornado” and also in the framework of the shared laboratory SIVALab between Renault, UTC and CNRS. The Tornado project had the goal of implementing an robotaxi service to perform an on-demand mobility service in the French town of



(a) (b)

Figure 1.2: (a) Demonstration of full autonomous driving and roundabout crossing in Rambouillet for the mid-line project demonstrations (November 2019). (b) Autonomous vehicles roundabout crossing during the Intelligent Vehicles Symposium demonstration session (June 2019).

Rambouillet. The project, which was carried out under the supervision of Renault, involved industrial and academia partners in order to study and develop different technical aspects of the mobility service. There was also a task on the acceptability of the service to the local population who were able to participate in the experiments.

During this PhD, several experiments and demonstrations have been carried out. In particular, the research team took part to several project demonstration sessions of the Tornado project. This has been useful to test and demonstrate the performance of the proposed algorithms on live tests. Figure 1.2a shows a moment of the demonstration given for the Tornado project mid-line demonstrations session. The system has been validated experimentally on an urban road closed to public traffic our roundabout crossing method. At that time, the main focus was on the navigation strategy and consequently no perception was used. To overcome this, V2V communication instead of vehicle on-board perception was used.

Moreover, we also took part in the 29th IEEE Intelligent Vehicles Symposium demonstration session in Paris on June 2019. Figure 1.2b shows the Intelligent Vehicles Symposium demonstration session where we proposed the same scenario of the first demonstration on the white car, while a Lidar-based obstacles detection algorithm was shown in the gray car.

2 Autonomous Navigation with HD Maps and Communication Systems: Basic Concepts and Tools

2.1 Introduction

The main objective of this chapter is to provide a general overview about the tools and concepts that will be used in this manuscript. This includes both the literature review on state-of-the-art methods and the description of the main tools that will be used in the experiments and tests to validate the proposed strategies.

This chapter is organized as follows. The first part of section 2.2 describes some methods that are used in cooperative Autonomous Driving (AD) vehicles navigation for collision avoidance and roundabout crossing, focusing the attention on roundabout navigation maneuvers, while the second part of section 2.2 contains a survey on the same topic but in the case of mixed traffic.

Section 2.3 proposes a general presentation on the HD map that is used in this work and its corresponding formalism, illustrating and comparing different methods for computing HD map based road models and their corresponding curvilinear coordinates.

Finally, section 2.4 introduces the main tools that we use to simulate a realistic traffic flow, while section 2.5 describes the experimental vehicles used for demonstration and data acquisition.

2.2 State of the Art

In this section, a survey about published contributions on different topics treated in this work regarding roundabout crossing is presented. In particular, we state the progress that have been made by theses works, highlighting also their drawbacks and the possible negative outcomes that such methods can have on our roundabout navigation use-case. This literature review is also useful to illustrate how the state-of-the-art knowledge has been exploited to design the navigation, perception and data fusion algorithms and how the existing methods have been adapted to the use-case of autonomous navigation in roundabouts.

2.2.1 Cooperative Navigation in Roundabouts

During the last decades, the research on autonomous driving systems has grown significantly. However, severe challenges for autonomous driving remain on some road sections that are complex for autonomous navigation, typically due to a lack of visibility. These are for instance tunnels, crossroads, roundabouts and railroad

crossings which are known to be the cause of most of the road accidents. Roundabouts are increasingly used in urban areas because they allow a safer approach than traditional crossroads with traffic lights. In fact, compared to traffic signals, the research of the Insurance Institute for Highways Safety (IIHS) indicates that roundabouts can lead to about a 90% reduction in fatalities and a 75% reduction in crashes with injuries [84]. Moreover, roundabouts guarantee also efficiency and are environmentally friendly, because a roundabout does not require a driver to stop unless there is traffic already in the circulating lanes. This can avoid to wait for a light to turn green or come to a complete stop if there is no traffic on the lanes. While it is common to find papers in the literature that propose strategies to handle a crossroad intersection [5, 23], even focused on cooperative driving control as [58], much fewer researches have been conducted regarding roundabouts. In [9], the authors proposed an algorithm that attempts to solve the roundabout crossing problem with the help of game theory, while some approaches to control vehicle inside a roundabout have been proposed in [81, 82]. Desaraju et al. [28] considered partial order techniques to develop a strategy in order to avoid a collision between two vehicles circulating into a roundabout. In these cases, the aspect that regards the control of the vehicles has mostly been considered, rather than the decision making part. Regarding the roundabout models, Rastelli et al. [81] used a very simple model based on a circle connected with some branches to model entries and exits, while the authors in [82] used Bézier curves to model them. One of the main advantages of virtual platooning [69] is to allow a single platooning control law to be adapted to several scenarios such as intersection crossing or lane merging. In this chapter, this concept is generalized to roundabout crossing.

2.2.2 Mixed-traffic Navigation in Roundabouts

In the near future, one can imagine that AD vehicles will co-exist with manually driven (MD) vehicles and other road users like bicycles or motorcycles. In this heterogeneous scenario, safe navigation has to be guaranteed in complex dynamic environments like in intersections, lane merging or roundabouts, where the risk of accident is one of the highest on public roads.

Concerning roundabouts navigation in mixed traffic, one way to ease this co-existence can be achieved by exploiting V2V communications to share to other traffic participants the AD vehicle intentions. In such a way, other vehicles can build driving plans considering explicitly the AD vehicle future behavior. However, this method assumes that all the road users are equipped with V2X devices, which is unrealistic in a short-term horizon.

Regarding the different approaches presented in the literature, many works have been done for intersection crossing with only AD vehicles [20, 57, 72, 78] and with priority constraints [26]. Moreover, there also exists some research works that aim at predicting drivers intentions and behaviors as in [41, 90]. Some of them propose to solve the intersection crossing problem with optimization techniques [72] or with model predictive control [64,89]. The main idea is to compute all the possible configurations of the system where a collision is unavoidable even if an emergency braking is executed and avoid them. Another way to address the problem is presented in [80]. Finally, the Autonomous Intersection Management Protocol (AIM) [19, 75] and virtual platoon methods [27, 69] have shown interesting properties.

In a mixed traffic environment with AD and MD vehicles (in the following an MD vehicle can be a car or any road user who uses the drivable space), avoidance sets

can be computed with the reachability theory [30, 55, 98]. The authors in [73, 74] have implemented a collaborative motion planning algorithm that cooperates with MD vehicles. Other works as in [105] rely on the MD vehicle behavior in the worst-case scenario (*e.g.*, the acceleration of an MD car approaching an intersection is considered to be the highest). In [37], the behavior of an MD vehicle is modeled with a mathematical representation that captures the human way of driving. Some tentacles-based motion planning techniques can also be found in the literature [92]. Such methods are often based on occupancy grids to estimate the free space in the environment [8] even in combination with HD maps [38].

In order to better understand vehicle behaviors in complex scenarios, we investigate in the literature the most important datasets that contain traffic flows information for autonomous driving navigation tasks (*e.g.*, collision avoidance, path planning, etc.). Several datasets can be found such as Common Road [6], ACFR [106] or INTERACTION [102]. In this study, it has been decided to focus our attention on the latter one because it provides highly dynamic traffic flow data in several different road scenarios. In particular, several roundabouts are present.

Concerning the roundabouts crossing problem, fewer works can be found in the literature, as we previously said in the preceeding section. Furthermore, another approach that involves an extra infrastructure layer to implement a time slot scheduling protocol can be found in [43]. Despite the fact that the authors proposed interesting and valid theoretical solutions, the experimental results have been tested only on some test-bed scenarios with either simulated vehicles or with simplified roundabout models. Moreover, experiments have been carried out with a limited traffic flow (only one adversarial vehicle), which is far from a full-scale experiment. If we look carefully at roundabouts, some similarities can be found with classical road merging or intersections. In particular, during the insertion maneuver, a roundabout is quite similar to a road merging where vehicles on the main road have higher priority. However, several fundamental differences shall be accounted for. First, in a roundabout the available space is limited w.r.t. a highway merging and consequently there is less time to plan a driving maneuver. Then, the traffic behavior in the roundabout ring is highly heterogeneous and hard to predict. This aspect is more visible if the roundabout has more than one lane. In particular, if no lane marking is present, it is not easy to distinguish between the innermost and outermost lanes in the roundabout. As a consequence, MD vehicles tend to have a highly irregular behavior, making the navigation maneuver in the roundabout challenging. Some examples of this behavior can be found in the roundabout recordings of the INTERACTION dataset [102]. Conversely, if one considers an intersection (T-intersection or lane merging), in general MD vehicles choose their lane before crossing the intersection or the merging. Another feature of roundabouts is that it can be difficult to predict which exit vehicles will take. To deal with this problem, it is then relevant to consider all possibilities by manipulating several virtual instances of the vehicles.

The concept of using virtual vehicles along the lanes of a HD map is very efficient to predict the dynamic situation in a roundabout and to control the longitudinal behavior of an AD vehicle. For this reason, this approach will be studied in details in this thesis.

2.3 High Definition Maps

When multiple vehicles share the driving space, it is important to properly represent their spatial positions with respect to each other. Euclidean coordinates are often not ideal for representing spatial relationships between road users. Curvilinear coordinates, on the other hand, are able to effectively encapsulate lane-level interactions between vehicles, for example in terms of conflicting trajectories w.r.t. the AD vehicle navigation corridor [76, 105]. The aim of this section is to introduce and explain the main elements concerning HD maps that will be exploited in the following chapters for both navigation and perception modules of our system. In particular, the attention will be focused on the computation of curvilinear coordinates w.r.t. a HD map.

2.3.1 High Definition Map Formalism

Nowadays there are a number of open source navigation maps (*e.g.*, OpenStreetMap, Vssim networks OpenDrive etc.). However, for AD navigation tasks, a HD map, with a higher level of detail and better accuracy, is needed.

The present work adopts a polylines formalism to represent the roads. This approach has been chosen in contrast to another well known standard that represents roads geometry with continuous curves as splines or polynomials. In particular, the road elements in a map are defined as follows:

- Node: a set N of geo-referenced 2D points used to mark the start and the end points of a part of a lane, in particular where two lanes split, merge or cross.
- Link: the portion of the lane between two nodes (the starting and ending nodes) that define the flow direction in the lane. The geometry of this part is represented as a polyline, which is a sequence of line segments. A link L_k composed of m_k line segments is described by $m_k + 1$ points

$$L_k = \left(p_k^{(0)}, p_k^{(1)}, \ldots, p_k^{(m_k-1)}, p_k^{(m_k)}\right), \tag{2.1}$$

  in which the first and last points are nodes, *i.e.*, $p_k^{(0)}, p_k^{(m_k)} \in N$. It is important to note that no crossing can occur within a link, but only at the starting or ending nodes. An example of this can be found in the map representation of Figure 2.1.
- Shape points: the 2D points used to model the geometrical shape of the lane *(i.e.*, the shape of the link) are called shape points. A node is also considered to be a shape point.
- Line segment: a line segment

$$l_k^{(i)} = \left(p_k^{(i)}, p_k^{(i+1)}\right), \tag{2.2}$$

  attached to a link L_k is composed of two consecutive shape points, and its length is defined as

$$\ell_k^{(i)} = \left\| p_k^{(i+1)} - p_k^{(i)} \right\|. \tag{2.3}$$

  A link L_k can be represented equivalently as an ordered sequence of m_k segments $L_k = \left(l_k^{(0)}, l_k^{(1)}, \ldots, l_k^{(m_k-1)}\right)$. The length of a link L_k is denoted

$$\mathcal{L}_k = \sum_{i=0}^{m_k-1} \ell_k^{(i)}. \tag{2.4}$$

To better illustrate this concept, Figure 2.1 shows the representation in terms of nodes and shape points for a given link. Another commonly used map representation is the *lanelet* representation [11], which uses the left and right bounds of the lane. In this case, a conversion into a center line representation is required beforehand, as in [48]. Those concepts will be explained better in the next section.

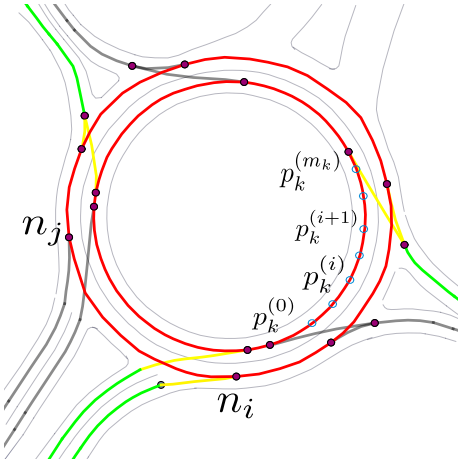


Figure 2.1: A roundabout with its HD map representation. The decision zones are green, the transition zones are yellow and the ring zone is red. The roundabout exits are gray. For a given link, the nodes and shape points are shown.

2.3.2 Curvilinear Coordinates Along HD Maps

The principle of curvilinear coordinates is to map an Euclidean pose, defined as $X = [x, y, \theta]^T$, where x and y represent the 2D position and θ is the heading angle of the AD vehicle in a Cartesian working frame onto a curvilinear pose with respect to a geometrical curve.

As it will be shown in the following part, there exists many curves that can be used to represent roads. Such curves can be stored in a map in a variety of formats including clothoids, splines, or polylines, and typically represent vehicles' nominal paths.

In many driving contexts, the center line of a lane is a sufficient approximation of the path that the vehicles follow. In the particular cases of overtaking and lane change, the AD vehicle can decide to replace its reference path by a new path computed by a planner. Let us define the curvilinear coordinates $[s, n, \psi]^T$, as illustrated in Figure 2.2, where s represents the AD vehicle curvilinear abscissa along the curve, n represents the signed lateral distance of the AD vehicle from the curve and ψ is the heading angle computed w.r.t. the curve.

To compute a curvilinear pose from a Cartesian pose, the first step is to implement a mapping between the center of the vehicle body frame and a point that lies on the curve. In order to chose such a point, a criterion must be established (e.g. the shortest point-to-point or point to curve distance). This process is called map-matching and Figure 2.2 illustrates such concept. As one can see, the point M is mapped into the point H on the curve via a map-matching process. Then, the curvilinear abscissa can be computed w.r.t. the curve. Considering also the tangent vector at point H, the signed lateral offset n and the relative angle ψ can be derived to compute the curvilinear pose. Notice that the component n has a negative value when M is at the right of the curve and it has a positive value when M is at its left. Every curve is oriented.

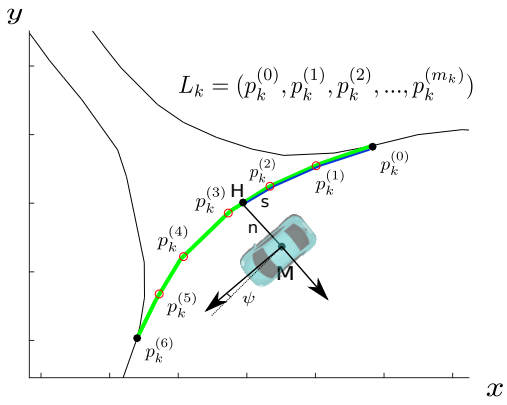


Figure 2.2: Curvilinear pose coordinates $[s\, n\, \psi]^T$ relative to a polyline of a HD map composed by the curvilinear abscissa s, the signed lateral distance n and the relative orientation ψ computed from a Cartesian pose $[x\, y\, \theta]^T$. The vehicle body frame is centered in point M.

As previously said in section 2.3.1, a HD map is a data structure that represents geometry and semantic of the urban environment at different layers. In its simplest form, a HD map can be provided as a sequence of points that represent the AD vehicle nominal path. Furthermore, this model can be refined to represent more complex driving situations. In a minimal representation, a HD map defined as the set: $\left\{p_k^{(i)} = [x_i,\, y_i]^T\right\}$ with $i \in 1, ..., n$ is considered. This set produces a set of oriented polyline segments that define the road orientation for traffic circulation. From this representation, it is possible to extract several different curves to represent the center of the lane. In this work, three types of model to represent the center of the lane have been considered. First, we consider the polylines model. This model is one of the simplest representations that we can exploit in such a context. The main idea of this approach is to connect two consecutive points $p_k^{(i)}$ and $p_k^{(i+1)}$ with an oriented segment. Such process is repeated for every couple of subsequent points of the map. To compute curvilinear coordinates with this new map representation, the map-matching is performed by searching the smallest distance among all the distances from point M to each segment of the polylines.

One other way to represent the centerlane consists in using a different curve representation instead of a polyline one. In particular, a polynomial representation is considered. This representation is obtained by fitting the set of points $\left\{p_k^{(i)} = [x_i,\, y_i]^T\right\}$. The goal is to obtain a smoother curve than polylines and to eliminate discontinuities at polyline vertex that appear when using polyline-based models. To perform map-matching, an optimization technique that minimizes the point-to-curve distance is exploited. In this work, we consider both B-spline and Hermite splines polynomials.

Finally, the lanelet model has been taken into account. This model exploits again a polyline-based framework to represent the centerlane of the road. However, this method exploits a non-Euclidean distance combined with a map-matching technique to overcome the discontinuities that arise when using polylines, providing a more continuous representation of the curvilinear coordinates. For further details about the three aforementioned techniques, the reader is invited to look at Appendix 7.3.

2.3.3 Map-matching Methods Comparison

Let us compare the road models and map-matching techniques presented before in order to understand their main advantages and drawbacks. A synthetic case study

is presented in order to illustrate discontinuity and non-linearity issues. Such a case study is useful to understand the effects that different map-matching and road models can produce during the AD vehicle navigation.

Let us consider a vehicle driving at a constant speed along a straight line. Figure 2.3 depicts such scenario. In particular, in Figure 2.3a, the dashed magenta line represents the AD vehicle trajectory. In order to build a navigation map to be used for validating the aforedescribed algorithm, a set of four control points is provided. The goal of this test is to compute the curvilinear coordinates of the vehicle with respect to the road map which is represented by the application of the different road models to the four control points. These points are illustrated by the red circles in Figure 2.3a.

On Figure 2.3a, we can see that if the polyline or the lanelet road models are used, the road path is represented by the set of line segments that connect subsequent control points (red line). On the contrary, regarding spline-based road models, we obtain a curve that passes through all the control points (green line) for the Hermite spline, while for the B-spline (blue line) it is provided as an interpolation of them.

2.3.3.1 Discontinuities and Non-linearities

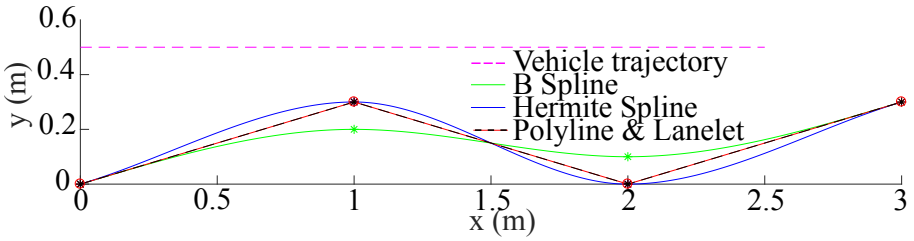
According to the different road models of Figure 2.3a, the resulting curvilinear coordinates computation, is depicted in Figure 2.3b, 2.3c and 2.3d. In particular, in Figure 2.3b, one can observe the resulting curvilinear abscissa for each model. It can be noticed that this quantity evolves in a different fashion for the four road models.

First of all, for the polyline model (red curve), one can clearly observe the presence of a stationary point when reaching the second control point at around one meter. In the neighborhood of this point, the value of the curvilinear abscissa remains constant while the vehicle keeps moving along the x-axis. After that, when it reaches the third control point, a discontinuity on the value of the curvilinear abscissa appears. This can be seen because of the presence of a jump in the red curve of Figure 2.3b.

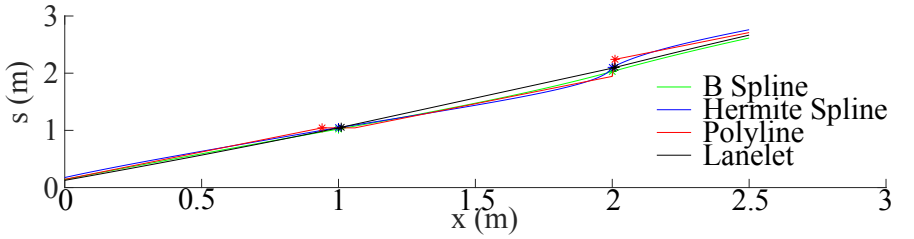
On the other hand, regarding the lanelet road-model (black curve), there is no discontinuity of the curvilinear abscissa value in none of the control points neighborhood. However, contrary to the polyline model, the curve between two control points is non-linear. This has the effect that, even though the vehicle is moving with constant velocity, the derivative of the curvilinear abscissa is not constant. This produces a false estimation of the driving speed of the vehicle. Moreover, the derivative of the curvilinear abscissa is not continuous in the neighborhood of the the control points, which means that, if we compute such quantity when reaching a control point, an infinite acceleration appears. The same phenomenon appears for the polyline models.

Finally, considering the result obtained by the use of the spline-based road models, one can observe that the continuity issues of the curvilinear abscissa and its derivative are not present. In particular, the curvilinear abscissa curves are continuous and differentiable for both the Hermite spline (blue curve) and the B-spline (green curve). However, a drawback of these two approaches is that a non-linearity appears in the zones between the control points. Such non-linearity is more severe than the one in the case of the lanelet model (this is particularly evident for the Hermite spline).

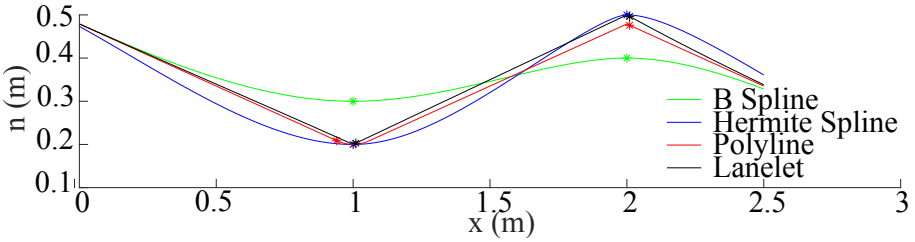
In Figure 2.3c, the resulting curvilinear ordinates are also reported. One can observe that the curvilinear ordinate is always continuous for the four methods. Nevertheless, the splines present a smoother behavior and, at the control points,



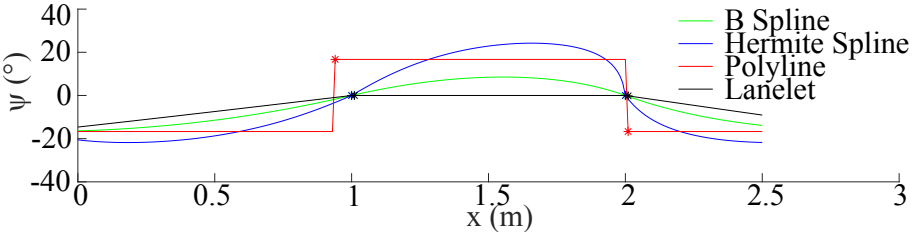
(a) Vehicle trajectory, B-Spline (in green), Hermite spline (in blue) and polylines used both for polylines and lanelet methods (in black and red).



(b) Curvilinear abscissa.



(c) Curvilinear ordinate.



(d) Curvilinear orientation.

Figure 2.3: Discontinuities on the curvilinear pose of a vehicle using B-Spline (in green), Hermite spline (in blue), polyline (in red) or lanelets (in black).

Table 2.1: Properties of the different road models.

	Polyline	Lanelet	Splines
s continuity		✓	✓
$\dot{s}$ continuity			✓
s linearity	✓		
n continuity	✓	✓	✓
$\dot{n}$ continuity			✓
n linearity	✓		
ψ continuity		✓	✓
$\dot{\psi}$ continuity			✓
ψ linearity	✓		
real-time	✓	✓	

a break appears in the curvilinear ordinate computed with the polyline and the lanelet. The derivative of the curvilinear ordinate of these two methods are also not continuous at the joint points.

Figure 2.3d illustrates the resulting curvilinear orientation. It is possible to see that the curvilinear orientation computed with the polyline jumps when the matched segment changes. The curvilinear orientation of the lanelet method is continuous but breaks appear at the control points and, as in the case of the curvilinear ordinate, the derivative is not continuous either. The splines methods always keep smooth trends with at least a C^2 continuity even at the joint points.

To sum up the results and the conclusions of the case-study, the different properties of the road models are summarized in Table 2.1.

2.3.3.2 Map-matching Discontinuity

In the previous case study, we have demonstrated that the curvilinear coordinates are at least continuous when using a lanelet-based and spline-based models to represent the driving corridor and for map-matching. However, the obtained results are actually valid only if the map-matching process is continuous itself. In other words, to provide continuous coordinates via a map-matching process, the vehicle must be map-matched to each polyline segments consecutively one after another. In the case of spline-based models, none of the portions of the curve between two consecutive control points must be skipped. However, there exists situations in which there is no guarantee that such constraint holds true. If we consider for example, cases where the vehicle is far away from the center of the lanes, the map-matching process may become discontinuous. This will inevitably lead to non-continuous curvilinear coordinates.

To visualize this concept, Figure. 2.4 illustrates the map-matched zones relative to each segment for the lanelet and splines methods. Because the tangents at the control points are the same for all these three methods, the boundaries between the different zones are exactly the same. One can see that if a vehicle drives on the upper part of the driving scenario, being far away from the road model, it will switch from the red zone directly to the blue zone, without going through the green zone. This will lead to a missed segment in the map-matching procedure. As a consequence of that, a non-negligible discontinuity in the curvilinear coordinates appears. The same reasoning holds both for AD and MD vehicles.

Nevertheless, within the context of an AD vehicle, it is reasonable to assume that

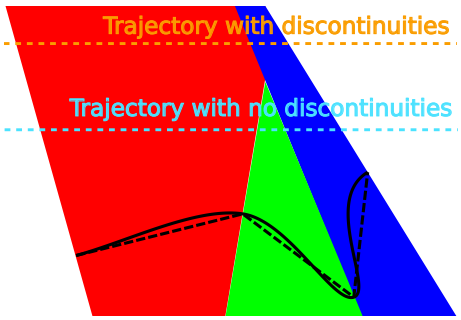


Figure 2.4: Map-matching discontinuities w.r.t. a polylines-based and a continuous road representation. The boundaries have been computed considering the lanelet map-matching and splines criteria. The red, green and blue zones correspond to geographic spaces for which the map-matching results to the first, second and third lane segments, respectively.

the vehicle always remains at a sufficiently short distance w.r.t. the center of the lanes. In such case, curvilinear coordinates using lanelet or spline road-models can remain continuous.

In the rest of this manuscript, the Lanelet formalism will be used to both implement the map-matching procedure and to compute curvilinear coordinates along the HD map polylines.

This work and its results have been published in the 20^{th} IEEE International Conference on Intelligent Transportation Systems [40].

2.4 Simulation with a Realistic Traffic Flow

In the simulation environment that has been developed for testing the navigation algorithms, the well known SUMO simulator has been integrated for handling the generation of an heterogeneous and random MD vehicles flow. However, in a simulated environment, it is difficult to simulate a traffic flow with realistic behaviors. Indeed, some driving maneuvers are often overly-simplified, as for example lane changes, and they do not correspond to realistic vehicle behaviors. The main reason is that, in general, motion and behavior models that traffic agents use in the simulated environments do not provide a sufficient level of details to model these kinds of maneuver in a general way. This is done to prevent the simulator to be overly-charged when a wide set of vehicles is simulated at the same time, for example for use-cases that need the presence of a wide set of cars moving together. On the other hand, a driving behavior model that is extremely focused on one single driving behavior, e.g. a lane change maneuver of an emergency brake can be used only for this specific purpose, producing limited results regarding the rest of all the possible maneuvers.

In the following, a technique that aims to re-create in a simulated environment the behavior of traffic agent recorded in a real traffic dataset is presented. In particular, we implemented a method to quantify the degree of adversariality of the obtained traffic flow w.r.t. the one in the datasets in order to both generate challenging driving scenarios for testing the navigation algorithms proposed later on Chapter 3 and to establish a ranking of the driving scenarios based on their complexity and interaction between road agents.

2.4.1 Traffic Simulation for Autonomous Vehicles

Simulation of the traffic environment is a technique widely used in relation to self-driving cars. In fact, this approach has several advantages. Simulations are safer, more efficient, and cheaper than live testing with real vehicles. With simulations it is also possible to generate a variety of different driving scenarios that would be hard to create using real road agents, thus reducing the risk of damage to material and injury to people.

Of course, in order for simulators to be meaningful when testing AD applications, a realistic simulation level w.r.t. the real world is essential.

In general, it is almost impossible to find a simulator that satisfies every single requirement for autonomous driving. However, there are a wide range of simulators that target individual aspects (traffic flow, sensor simulation, etc.).

In this work, we focus on traffic flow simulators. Simulators concerned with traffic flow can be separated into two main types, namely *macroscopic* and *microscopic* traffic simulators. Macroscopic simulators are mainly used to study large-scale flow problems, relating to road capacities and bottlenecks over a large complex road network. In these simulators, the dynamics of individual traffic agents are quite simple, and sometimes there is an absence of information relating to single traffic agents. The level of details is kept low to allow the simulator to perform simulations involving hundreds of vehicles and with a reasonable computation time. Macroscopic simulators are therefore not suitable for AD navigation because interactions between the vehicles and driving maneuvers (*e.g.*, lane changes) are overly simplified. As a consequence, they do not provide a sufficient level of detail to be used for testing a navigation algorithm.

Microscopic simulators, on the other hand, focus on the characteristics of single road agents rather than global vehicle flow. The description level of every simulated road agent must be high and detailed. Longitudinal and lateral motion of vehicles can be obtained via a number of built-in driving models customizable with a large set of driving parameters. For some simulators, *e.g.*, SUMO, it is also possible to parameterize some complex maneuvers such as lane changes, for example modeling the duration of the maneuver and the overtaking strategy.

For this work, it has been decided to use the well-known SUMO simulator [51], widely used for microscopic traffic simulation in the area of collision avoidance [32, 49]. We refer interested readers to [32] for more technical details about it.

Although simulation performance is becoming increasingly precise, it is important to be aware that a gap still exists between simulated maneuvers and those performed in the real world. However, state-of-the-art simulation technology is helping to narrow that gap, with regular new releases of the simulators.

2.4.2 Vehicles Flow Generation

This section proposes a method to reproduce a realistic vehicle flow in a simulated environment and to quantify its degee of adversariality w.r.t. the AD vehicle. To do so, we use data contained into the real traffic dataset INTERACTION. As stated previously, this dataset has been chosen because it provides data for highly interactive road users, including situations where road users have adversarial motion behaviors. With the term "adversarial" we mean a non-cooperative navigation behavior that has the goal of fulfilling the single vehicle behavior, rather than optimize the collective navigation of all the road users. As a consequence, sometimes adversarial vehicles tend to have aggressive behaviors and they may not respect traffic rules.

Traffic flow data recordings at a microscopic level in a dense traffic flow situation are provided for every roundabout in the dataset.

In order to compare the performance of the generated flow in the simulator w.r.t. the original flow, a metric is required. This metric must be able to capture the degree of interaction between road users, especially in complex driving situations (close navigation in parallel lanes, overtaking, intersection insertions without markings, etc.).

The authors in [102] propose to use the $\Delta TTIC_{min}$ (minimum Time-To-Imminent-Conflict-point) difference metric for an interactive pair of vehicles [93]. Following [102], we define an interactive pair of vehicles as a pair of vehicles that have at least one common point in their respective reference paths.

For a given interactive pair of vehicles, the corresponding $\Delta TTIC_{min}$ can be computed as

$$\Delta TTIC_{min} = \min_{t \in [T_{start}, T_{end}]} (TTIC_1^t - TTIC_2^t), \tag{2.5}$$

where $TTIC_i^t = \triangle d_i^t / v_i^t$, and $i = 1, 2$, is the traveling time to the conflict point of each vehicle in the interactive pair and the time interval $[T_{start}, T_{end}]$ represents a time window where two vehicles interact. Note that $\triangle d_i^t$ is the distance from the vehicle position to the conflict point in a Frénet frame, which is equivalent to the absolute value of the curvilinear abscissa w.r.t. to a conflicting node, as defined in Equation (3.2). When $\Delta TTIC_{min} \leqslant 3$ s, there is deemed to be an interaction between the vehicles. Moreover, an interaction between two vehicles is termed *intense* if $\Delta TTIC_{min} \leqslant 1$ s. In this study the focus is put on the data from the "USA_Roundabout_FT" scenario, because this is the most interactive two-lane roundabout, with the highest number of vehicles and recorded sequences.

We will now explain how SUMO can be used to generate a flow of vehicles based on the INTERACTION traffic data with the same degree of interaction in terms of $\Delta TTIC_{min}$. This has several advantages. It allows us to quantify the performance of the algorithm w.r.t. the interaction degree of the traffic scenario. It can give a vehicle flow with the same degree of interaction as on a different roundabout test-bed. A simulated vehicle flow in SUMO can react dynamically to the ego vehicle decisions (*e.g.*, braking when the insertion maneuver is too aggressive), while this is not the case for a recorded flow dataset.

Using the curvilinear formalism introduced in section 2.3.1, an algorithm to compute common intersection points between vehicle trajectories has been implemented. In this case, trajectories are encoded as a list of links that a vehicle follows on the road map, and consequently the intersection point is one node of the map. Unfortunately, neither vehicle trajectories nor a lane-level high-definition map are present in the INTERACTION dataset, and they therefore need to be computed both off line.

To do so, a HD map representation of the "USA_Roundabout_FT" has been computed starting from its *lanelet2* [76] representation provided with the dataset interface. The main idea behind this step is to convert the *lanelet2* representation of street borders to a centerline representation using our curvilinear formalism. A Voronoi distances algorithm has been used to compute the set of center lines, that is to say equidistant points from lane borders. However, this procedure is not completely automatic and can be error-prone where roads intersect and merge. Some manual corrections were therefore done off line at the end of this step.

Although the "USA_Roundabout_FT" has more than one lane, we represent it with only one lane in the HD map. This is to better capture the interaction between

vehicles inside the roundabout ring, assigning vehicles navigating on multiple lanes to the same portion of the map, even where vehicle trajectories do not overlap. This is a reasonable strategy given that navigation in parallel lanes implies a possible implicit interaction that should be taken into account (we cannot know in advance the intentions of the vehicles).

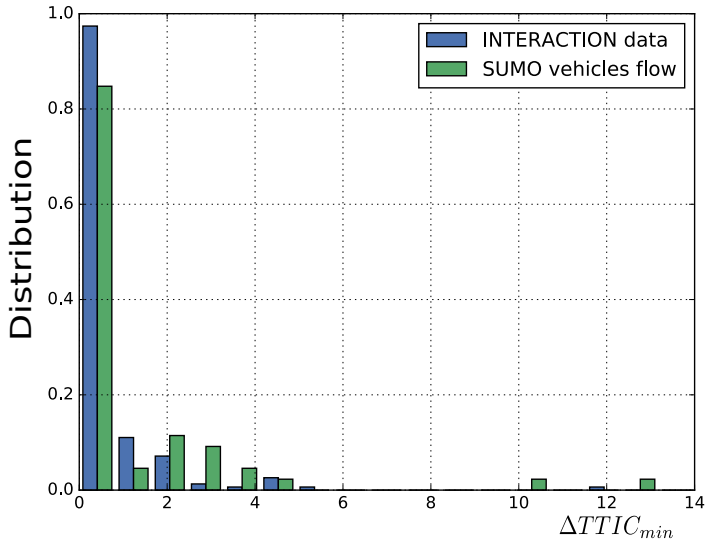


Figure 2.5: Comparison between the $\Delta TTIC_{min}$ distribution generated from the same sequence of the USA_Roundabout_FT scenario of the INTERACTION dataset (blue) and the one obtained from the traffic flow simulated by the resulting traffic flow simulation in SUMO (green). One can notice that an highly dynamic behavior ($\Delta TTIC_{min} \leqslant 1$ s) is present in both scenarios.

To compute a reference trajectory for the vehicles w.r.t. the HD map, we have implemented a map-matching procedure via the INTERACTION dataset animation tool. From the raw data on trajectories this procedure directly computes the corresponding curvilinear position on the HD map and lists all the map links that a vehicle traverses.

Once the intersection points between every interactive pair have been found, we use the HD map to compute the curvilinear distances $\triangle d_i^t$. Technical details about this inter-distances computation can be found in section 3.3.6.

As for the input data to SUMO, for every track of the "USA_Roundabout_FT" the arrival times, arrival speeds, inter-distances w.r.t. the vehicle ahead and behind have been computed. The empirical distribution of these observed data was estimated and used to randomly create vehicles with the same properties as their real-world counterparts. Finally, the speed of each vehicle was bounded by the speed limit provided by the road shape and the route attribute of each vehicle object was chosen randomly, with more weight given to routes intersecting the AD vehicle's route.

The output of the system is a traffic flow with the $\Delta TTIC_{min}$ distribution similar to that computed using the real traffic data. Figure 2.5 illustrates the output of the simulation for a recorded traffic sequence from INTERACTION. Highly dynamic behavior ($\Delta TTIC_{min} \leqslant 1$ s) is present in both scenarios.

The high degree of interaction between interactive pairs comes mainly from the double lane of the roundabout ring. Where two or more vehicles are traveling very

close to each other in parallel lanes in the roundabout ring, there is a high volume of interaction between them.

This method has been published in the IEEE Intelligent Vehicles Symposium conference [67].

2.4.3 Hybrid Simulation Environment

After having discussed in details the method for simulating a realistic and highly interactive traffic flow in our simulated environment, we present in this chapter another tool that allows to move our simulation a step further closer to the reality.

Figure 2.6: Hybrid simulation experiment on the circuit Seville. On the right of the figure, the MD traffic generation in SUMO (yellow cars) and the AD vehicle (gray car, in the red circle). On the left, the corresponding intelligent vehicle framework with the same obstacles (yellow rectangles). As all the MD vehicles are simulated, they do not appear in the AD vehicle context camera

Let us consider again the simulator presented in section 2.4.2. Such a simulator is composed of two main components: the SUMO-based simulator, for generating an MD traffic flow and the ROS-based simulator, for simulating the motion of the AD vehicle.

In this architecture, it is possible to replace the ROS-based simulator by connecting directly the SUMO-based simulator real vehicles. This is possible because the system has been designed to be modular w.r.t. different components, and both the intelligent vehicles architecture and the SUMO-based simulators frameworks share the same middle ware (ROS). In other words, it is possible to simulate the presence of virtual obstacles in the navigation space of the intelligent vehicle. Such vehicles are generated by SUMO according to the method described in section 2.4.2 and sent to the intelligent vehicle on-board system.

However, to generate a consistent scenario, both the SUMO-based simulator and the intelligent vehicle have to share the same HD map of the driving scenario.

Once this step is done, the intelligent vehicle can be programmed to perform a certain mission (e.g. going through a roundabout) and, as soon as a virtual obstacle appears, the intelligent vehicle will modify its behavior according to the driving situation, treating the virtual obstacle as if it was a real one.

To better understand such situation, Figure 2.6 depicts an example of this technique. As one can see, on the right part of the figure, we have the SUMO with the HD map representation of the experimental circuit Seville (section 2.5.5.1). Notice

that some roundabout branches have been artificially added to generate a more complex MD vehicles flow. On the other hand, it is possible to see the HD map-based AD vehicle navigation space of the same driving scenario, with the corresponding MD vehicles generated in SUMO. Notice that these vehicles do not appear in the intelligent vehicle context camera.

2.5 Experimental Setup

In this last section of the chapter, we present the experimental platform of the Heudiasyc laboratory that have been used in this thesis. In particular, the experimental platform has been exploited both to test the navigation algorithms discussed in chapter 3, and to acquire real-traffic datasets for testing our perception system described in Appendix 4.

Those datasets have been recorded to propose a common benchmark from different shared perception use cases. The detailed explanation of the context and of the different scenarios of these datasets is given in chapter 7.

Those data have been recorded in both the cities of Compiègne and Rambouillet, in France between 2018 and 2020, under different traffic and weather conditions.

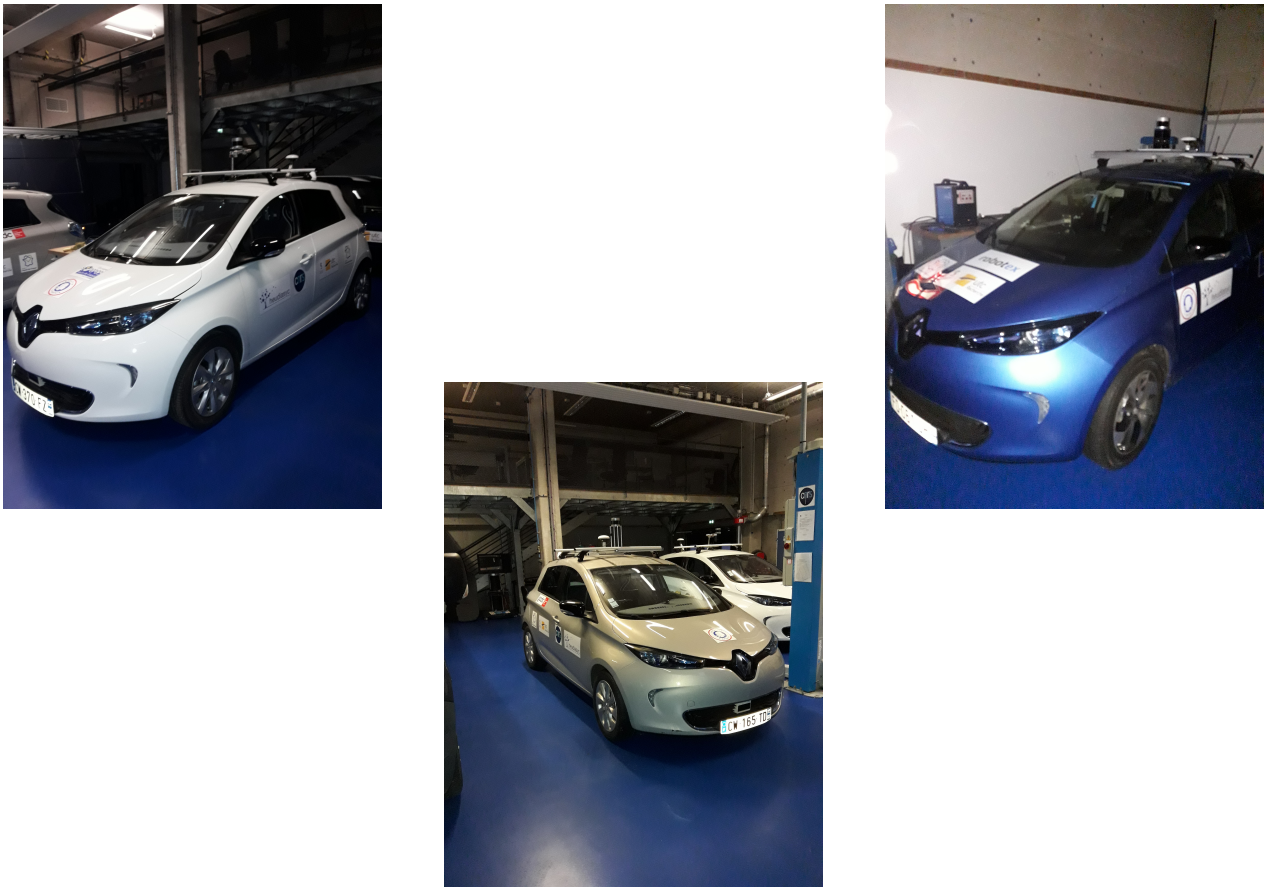
2.5.1 Experimental Vehicles Platform

The Heudiasyc laboratory owns a research vehicles platform with several prototypes of intelligent vehicles. Such vehicles can be used both to test autonomous vehicle driving, and to collect data from real traffic environments.

Figure 2.7 illustrates the three Renault ZOEs of the platform. Vehicles illustrated in Figures 2.7a and 2.7b are part of the ROBOTEX (ANR-10EQPX-44-01) project and are fully controllable by an on-board computer via a micro-autobox device that communicates directly with the car engine. In particular, it is possible to control the vehicle acceleration and brake by generating a torque, and the vehicle's steering wheel by generating a steering wheel angle. The other car, shown in Figure 2.7c, has not been modified to be controlled in autonomous mode.

Regarding the onboard sensors, all the three vehicles are equipped with a different set of sensors to be used both in experiments and for data recording. A list of the most relevant sensors is provided hereafter:

- Front and rear multi-layer LiDARS
- A 360° 32 Layers LiDAR on the vehicle's roof
- Several On-board cameras
- An intelligent Mobileye camera
- Several GNSS localization receivers (with the possibility of exploiting Real Time Kinematics corrections in real time and to do PPK for very accurate positioning in post-processing)
- 802.11p radio modems for both V2V and V2I communication



(a) (b) (c)

Figure 2.7: The three vehicles of the experimental platform of the Heudiasyc laboratory. Figure 2.7a and 2.7b depict the two vehicles equipped with the control system, respectively called the Zoe Grey and the Zoe White, while Figure 2.7c shows the vehicle used for data recording, called the Zoe blue.

2.5.2 On-board Sensors

We describe here the main on-board sensors involved in this work. In particular, the attention will be focused on the localization system that is used to provide a ground truth localization for the AD vehicle and the 32 Layers LiDAR sensor that we use for perception tasks.

2.5.2.1 Localization

To obtain an accurate and robust localization for the AD vehicle, a NovAtel SPAN CPT GNSS receiver with an Inertial Measurement Unit (IMU) is used. Such a system provides an accurate and reliable localization by the combination of GNSS positioning and information from accelerometers and gyrometers.

Moreover, localization accuracy is improved by exploiting Real-Time Kinematic (RTK) corrections in real-time. Such corrections are provided via a fixed receiver. An accurate knowledge of the antenna of this receiver is required. In this case, an antenna has been installed on the roof of a building of the UTC campus. This antenna sends the RTK corrections to the on-board receiver installed on the AD vehicle providing a centimeter level of accuracy in localization. The uncertainty on positioning provided by the sensor is around 10 centimeters on latitude and longitude and around 3°/4° on yaw angle orientation in the worst case. However, for data acquisitions performed away from the campus area, either NRTK or post-processing RTK corrections have been used.

This system has been used for several purposes. First, to provide an accurate localization to the AD vehicle during experiments and real-time demonstrations on the experimental site of the campus. Second, to provide a ground truth localization for the recorded datasets. In this case, to have a full range cover on all the zones, we exploit a post-processed PPK instead of RTK corrections in real-time. In such cases, the three vehicles of the platform have been used for recording (details are available in Appendix 7.2). For every vehicle, a localization system provides the localization to be used either as the AD localization source or as the ground truth that provides the exact localization for a detected road obstacle.

The coordinates provided by the localization system are converted from a global reference frame (e.g. WGS 84) into an East North Up (ENU) frame coordinates. The ENU frame is obtained by approximating the spherical earth's surface with a tangent plan centered at a fixed point. In our case, for each scenario of the datasets, a point close to such driving environment has been chosen. In the rest of the manuscript, the localization of the AD ego vehicle in the world reference frame will be indicated as:

$$ {}^{w}X_e = \begin{bmatrix} {}^{w}x_e \\ {}^{w}y_e \\ {}^{w}\theta_e \end{bmatrix} \tag{2.6} $$

Where the superscript index w indicates the ENU frame. Notice that in this work, only a 2D pose is computed and the height coordinate is disregarded.

2.5.2.2 360° LiDAR Sensor

To perceive the AD vehicle's surrounding obstacles, a 360° 32 beams Velodyne VLP32-C LiDAR (Light Detection and Ranging) has been used. This sensor scans the driving environment revolving around its center, providing a 3D environment representation in the format of a dense point cloud.

To build such a representation, the LiDAR sensor performs a non-contact distance measurement by emitting a series of light pulses and measuring the impact distance and the impact position along the X, Y and Z-axes of the series of points obtained when the light beams hit an object. The travel distance of each pulse is obtained by considering the pulse time-of-flight measurement. This time-of-flight is computed by measuring the time difference between the time instant when the pulse is emitted from the sensor and the time instant when the pulse returns to the sensor.

The choice of using of a laser pulse that moves at the speed of light allows to detect the shortest distance between the sensor and a certain detected object. This prevents to consider any other possible deflection. However, other different problems arise, such as the impossibility to see through detected objects. As a consequence, this may generate occlusions and blind spots in cluttered traffic environments. Another issue that may arise is the phenomenon of multi-echoes. In this situation we have multiple returning pulses that correspond to a single emitted light beam. This can be caused by the presence of elements that partially reflect the laser beam, as for example transparent objects or natural phenomena like rain, fog or dust.

The Velodyne VLP32-C is a particular LiDAR sensor that comes with 32 laser beams disposed along the vertical axis of the sensor, with a vertical field of view between $+15°$ and $-25°$. Moreover, it has an horizontal field of view of $360°$ and it can range up to 200 meters. Such beams are emitted simultaneously and the sensor revolves around its barycenter with a frequency of 10Hz (600 RPM). This leads to a scanning with an azimuth resolution of $0.2°$.

The scanned points are stored into a point cloud data structure. A point cloud is provided for every $360°$ revolution of the sensor. When a pulse is not reflected back or it does not hit any object, no point is obtained. In general, each scanning sequence provides around 60000 points. Each data point belonging to the point cloud is acquired and provided in spherical coordinates. For each point, the radius r, the elevation ω and the azimuth α are computed. Such quantities are respectively obtained by radial impact distance, the vertical and horizontal angles of the laser beam that generated the specific impulse. These spherical coordinates are converted into Cartesian coordinates in the sensor frame by the Velodyne driver module by computing the following transformation:

$$ {}^{v}P = \begin{bmatrix} {}^{v}x \\ {}^{v}y \\ {}^{v}z \end{bmatrix} = \begin{bmatrix} r\cos(\omega)\sin(\alpha) \\ r\cos(\omega)\cos(\alpha) \\ r\sin(\omega) \end{bmatrix} \tag{2.7} $$

Where the angle α represents the azimuth angle, r the radius and ω the elevation angle.

2.5.3 V2X Communication

To implement I2V communication between the AD vehicle and the intelligent infrastructure during the tests, the V2X technologies provided by Lacroix-City group has been used. We have used an On Board Unit (OBU) and a Road Side Unit (RSU) to achieve I2V communication. Such devices are able to encode, broadcast and decode messages in the main formats of the ETSI standard (Appendix 7.1). In this specific case, the Cooperative Perception Message (CPM) for broadcasting the results of perceived information between the intelligent infrastructure and the AD vehicle has been used. A detailed explanation of the exchanged information between the two systems is discussed in Appendix 7.1. Furthermore, the devices are equipped with a

(a) (b)

Figure 2.8: Equipment used to implement I2V communication. Figure 2.8a: On-board Unit (OBU). Figure 2.8b: Road Side Unit (RSU).

GNSS receiver that allows to synchronize the different clocks in the same time basis via a synchronization protocol. This step is crucial to exploit information in the different systems. Figures 2.8a and 2.8b, show respectively the OBU and the RSU used during our experiments.

In order to record our own cooperative perception datasets, another model of Cohda 802.11p radio modems has been also used.

2.5.4 Intelligent Infrastructure

One of the objectives of this work is to study the benefits that an intelligent infrastructure can provided to enhance safety during complex driving maneuvers. In this use-case, it has been decided to exploit an intelligent camera to monitor road traffic in a roundabout (Figure 2.9). Such a camera is able to process the acquired images in real-time and to broadcast the obtained results to the AD vehicle. To do so, the camera has been installed in a way that it can provide additional information about incoming MD vehicles on the left entering branch of the roundabout w.r.t. the AD vehicle trajectory.

This part has been developed in the context of the Tornado project in collaboration with the Université Gustave Eiffel.

In order to provide exploitable perception information, the camera raw image are sent to a computer via Ethernet. The received images are then treated to extract bounding boxes of road users and their estimated speeds. In order to extract the aforementioned quantities, an implementation of the YOLO V3 [45] detection algorithm has been used. Such a technique is capable to capture information as the class of a detected object and the object bounding box in the image plan. The dimension of the object are then inferred by combining the class information and the object bounding box in the image plan.

After that, the bounding boxes are projected into the word frame via geometric transformation and exploiting the HD map-based road geometry representation. Notice that this procedure requires a non-trivial transformation. Details about this process will be given in chapter 5.

Finally, the perceived information is encoded into a CPM object and sent periodically to the AD vehicle with the OBU.

The overall rate of this treatment is around 10Hz. Notice that, as in the case of V2X communication, the camera's clock is also synchronized at the same time as the rest of the system.

Figure 2.9: A surveillance camera used for monitoring road traffic in the roundabout scenario.

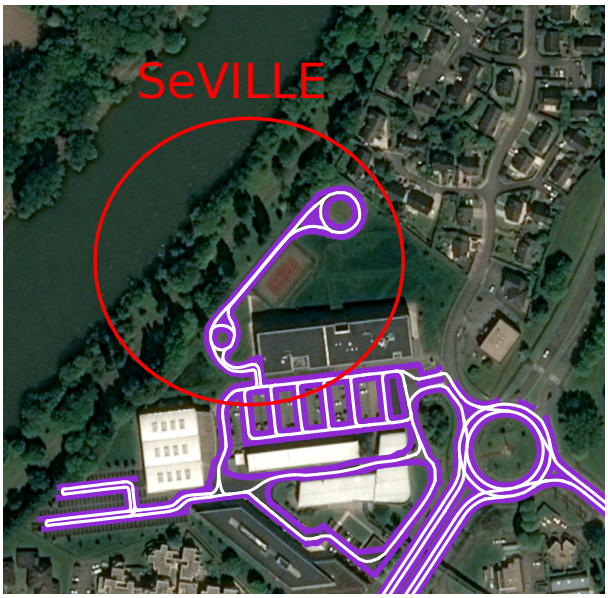


Figure 2.10: The HD map representation of the UTC campus. In the red circle, one can see the SeVILLE experimental site.

2.5.5 Test Scenarios

In this last part of the chapter, the main scenarios where the experiments and tests with the intelligent vehicles took place will be presented. Moreover, some of these sites have also been used to collect datasets for validating the methods proposed in chapter 4 and chapter 5. A further description of these datasets is given in Appendix 7.1.

2.5.5.1 Test Site: SEVILLE

The first testing scenario that we describe is the experimental circuit SeViLLE. This site is an outdoor environment located at the Heudiasyc Laboratory in the city of Compiègne. This site is closed to public traffic and it is used as a test-bed for the intelligent vehicles of the laboratory platform.

Figure 2.10 illustrates the map of the Heudiasyc Laboratory campus and the corresponding experimental circuit. Such a circuit is composed of two little roundabouts and a road (that can be traveled in both directions) that joins them.

In this work, the circuit has been useful to test the navigation algorithms both in a hybrid simulation context (section 2.4.3) and in a real experiment with the other vehicles of the platform.

(a) (b)

Figure 2.11: The city of Compiègne (France) with its corresponding HD map. The area (a) represents The Guy Dénielou roundabout in which data collection has been performed, while area (b) represents the bridge where the intelligent infrastructure has been installed.

2.5.5.2 Compiègne Scenario

Another experimental driving scenario that has been considered in this work is the city of Compiègne. It includes several zones of the municipality. These zones also have a HD map representation that contains several roundabouts and other complex traffic zones. Figure 2.11a provides a global overview of the town with the corresponding HD map.

The most used part of the city was the Guy Dénielou roundabout that has been exploited for data acquisitions. This roundabout is situated next to the Heudiasyc laboratory. Figure 2.11a illustrates this experimental scenario, while Figure 2.11b depicts the roundabout that has been used for the tests.

2.5.5.3 Rambouillet Scenario

The Tornado project took place in the city of Rambouillet. For this reason, several experiments have been also carried out in this site. Figure 2.12a shows a bird-eye view of the whole city. The aim of the autonomous mobility service was to link the different parts of the municipality. Figure 2.12a shows the whole municipality of Rambouillet and its corresponding HD map. As one can see, such scenario includes both urban environments and countryside roads.

In order to test cooperative navigation of AD vehicles on roundabouts, we have installed an infrastructure-based system in the roundabout (a) of Figure 2.12b. This choice has been done because we considered that this roundabout is the most challenging both in terms of reduced visibility and traffic load.

2.6 Conclusion

In this chapter, we have presented an overview of the main tools that will be used in the rest of the manuscript. For each research topics of this work, a state-of-the-art review has been carried out. Then, a HD-map based formalism for computing curvilinear coordinates w.r.t. polylines has been introduced. A comparison of the main methods for modeling a reference trajectory from a set of given points has been performed. In particular, the comparison has been carried out looking for the continuity of the computed curvilinear coordinates and the real-time implementation

(a) (b)

Figure 2.12: The city of Rambouillet (France) and its corresponding HD map representation. In Figure 2.12b, the urban scenario with some roundabouts. Roundabout represented by area (a) has been chosen to test the infrastructure-based system, while roundabout represented by area (c) has been used to collect some data. Finally, the path represented by area (c) has been used to test some autonomous driving use-cases.

of the methods. The results show that, in order to achieve our objectives, the lanelet technique is the best one. Indeed, this technique provides a continuity in the obtained curvilinear coordinates and it can be computed in a reasonably low amount of time. The concept of curvilinear coordinates along with HD map is crucial to obtain a consistent representation of the curvilinear inter-distances between different road agents, as it will be shown in the next chapter. If the continuity is not verified, some unwanted behavior in the inter-distances computation can appear, leading to a misleading representation of the interaction between road agents.

We have also introduced a novel approach for enhancing the simulation capabilities of the well-known traffic simulator SUMO in the case of a dense traffic flow. In this specific case, data recorded from several dense traffic scenarios all over the world have been exploited to re-create the same kind of traffic scenarios in SUMO. To compare the degree of interaction between these traffic flows, a metric has been introduced. It is opinion of the authors this kind of approach allows to simulate a more interactive and realistic traffic for simulation purposes. Such kind of approach also allows to capture interactions between vehicles in dense traffic jams, for example nudging and the non respect of traffic rules, which must be included in a navigation test-bed in order to provide a set of use-cases as general as possible.

Finally, we have presented the main systems and sensors that we will exploit during the experiments and also provide a description of the different scenarios where tests and datasets recordings took place.

3 Map-based Navigation Methods to Cross Roundabouts Safely

3.1 Introduction

Despite the tremendous growth of research regarding fully autonomous driving vehicles (AD) in the past few years, many safety critical scenarios, such as the crossing of roundabouts, are still open issues. Vehicle-to-vehicle and vehicle-to-infrastructure communications offer an appealing solution to handle such situations in a cooperative way. In this chapter, we propose to adapt the concept of virtual platooning to the roundabout crossing use-case. This idea allows a single navigation strategy to handle complex scenarios such as intersection and roundabout crossings in a general and scalable fashion. First, a demonstration of the efficacy of this algorithm in a case where only communicating vehicles are present in the driving scenario is given. Then, this approach is generalized not only to communicating AD vehicles, but also to manually driven (MD) communicating vehicles or even non-communicating ones.

In the second part of the chapter, an extension of the aforementioned navigation strategy is proposed to deal with navigation scenarios with one self-driving car among a flow of non-communicating and non-cooperative MD vehicles. One of the major challenges of this driving scenario is the safe navigation of AD vehicles on roads open to public traffic. Indeed, behaviors and intentions of MD vehicles are hard to predict and understand.

The presented approach relies on High-Definition (HD) maps with a lane level description which allows to predict the future situation thanks to the concept of virtual vehicles. This method handles safely collision avoidance and guarantees that no priority constraint is violated during the insertion maneuver without being overly cautious. The performance is evaluated with the SUMO simulation framework. A highly interactive vehicles flow has been generated using real data from the INTERACTION dataset, according to the method discussed in the previous chapter. We also propose strategies to extend the algorithm to multi-lane roundabouts and report how these extensions behave in terms of safety and traffic flow. Finally, the feasibility of this approach has been shown performing real tests carried out with an experimental AD vehicle on a test circuit in real time. The results show that this approach is easy to integrate into an embedded system and that it allows roundabouts to be crossed with a high level of safety. This chapter is organized as follows: in section 3.2, it is presented the roundabout crossing algorithm in the case of a single-lane roundabout, with communicating and cooperative self-driving cars, with no uncertainties on vehicles localization and no priority constraints. Section 3.3 presents the algorithm in the most generic case where all the aforementioned simplifications have been progressively removed. Finally, section 3.4 illustrates the main experimental results and in section 3.5 we discuss the main conclusions and future perspectives about the presented work.

3.2 Single-lane Roundabout Crossing with Cooperative vehicles

The aim of this first part is to illustrate the main concepts of a navigation method in a scenario with only cooperative and communicating vehicles. Moreover, in this first part of the work, priority constraints between vehicles in the roundabout are not still considered. This choice is motivated by the fact that if all the vehicles are autonomous and cooperative, there is no need to impose priority rules to regulate traffic flow in a roundabout. This assumption will be removed in the next section, where the extension of the approach to MD vehicles will be discussed. In this work, the word vehicle is used to represent several different kinds of road users, as for example: cars, trucks, buses, bicycles, etc. This choice has been done to underline the capability of the algorithm to work with any different kind of road agent.

3.2.1 Curvilinear Signed Inter-distance

The first step of the curvilinear representation is to get the lane followed by the vehicles. For an AD vehicle, it is simply given by the path planning module in the form of a list $N{=}(L_1, L_2, ...)$ of links to follow. In contrast, the driving lanes of the MD vehicles need to be estimated by a map-matching procedure. It is well known that map-matching ambiguities may arise when a road splits or when the vehicle changes lanes. For this, a map-matching process is exploited. This process is achieved using the method that will be described in section 3.2.3 [66] where multiple candidate lanes can be simultaneously occupied by creating virtual instances of the MD vehicle, one for each lane possibly occupied. This technique will be detailed in section 3.3.3.

According to the notation introduced in the previous chapter, the curvilinear abscissa s_{L_k} of a vehicle lying on the link L_k and map-matched to its i-th segment $l_k^{(i)}$ is computed w.r.t. the starting node $p_k^{(0)}$ of L_k as follows:

$$s_{L_k} = \sum_{j=0}^{i-1} \ell_k^{(j)} + \lambda \ell_k^{(i)}, \tag{3.1}$$

where $\lambda \in [0, 1]$ is a parameter to model the position along the map-matched segment $l_k^{(i)}$. In the rest of the chapter, s_{L_k} will simply be noted s when there is no ambiguity. Practical computation of the curvilinear coordinates can be found in [40].

This representation is particularly useful to do virtual platooning for safely crossing an intersection with a vehicle driving on another road [69]. Given the path of the vehicle $N = (L_k, L_{k+1}, \ldots)$, the curvilinear abscissa to any node from the path, for example, the ending node of the link L_n, is computed as

$$s_n = \sum_{j=k}^{n} \mathcal{L}_j - s_{L_k}. \tag{3.2}$$

Equation (3.1) computes the curvilinear abscissa of a vehicle measured w.r.t. the start node of the current link L_k, while in equation (3.2) the quantity s_n is measured from the ending node n of link L_n backward to s_{L_k}. Note that s_n is negative and increases as s_{L_k} approaches the node n.

Let us consider two vehicles V_i and V_j with paths N_i and N_j, respectively. If the two paths intersect, the curvilinear abscissa can be used to compute the relative

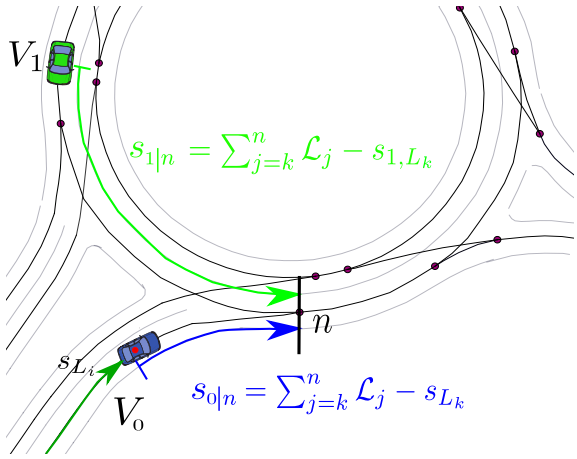


Figure 3.1: The curvilinear abscissa w.r.t. a given link and the relative distance computation to the intersection point. The blue arrow pointing towards the intersection point indicates the sign of the distance.

virtual gap between the two vehicles. Suppose that N_i and N_j intersect at a given node n, and let $s_{i|n}$ and $s_{j|n}$ be the curvilinear abscissa of V_i and V_j w.r.t. this node n as defined in equation (3.2). We define the virtual curvilinear signed inter-distance between V_i and V_j as follows:

$$d_{i,j|n} = -d_{j,i|n} = s_{j|n} - s_{i|n}. \tag{3.3}$$

Note that $d_{i,j|n}$ is a signed inter-distance where $d_{i,j|n} > 0$ means that V_i is closer to n than V_j, *i.e.*, V_i is virtually ahead of V_j. Figure 3.1 illustrates the computation of the curvilinear abscissa and distances.

3.2.2 Adaptation of the Virtual Platooning Technique

In this section, the main concepts of inter-distances computation and map-based curvilinear coordinates seen in chapter 2 are applied in the navigation algorithm for the case of a roundabout crossing for cooperative AD vehicles that exchange their pose and future path. In particular, an adaption of the virtual platooning technique to ensure a safe roundabout crossing maneuver will be explained in the following part of the chapter.

Suppose an AD vehicle V_i needs to cross a roundabout in the presence of other AD vehicles V_1, $V_2, \ldots,$ V_n in the roundabout or approaching it. Let us first assume for simplification purposes that the vehicle is alone on its way to the roundabout. The first idea is to find a leader vehicle V_j, if it exists, w.r.t. which the vehicle V_i needs to do virtual platooning. V_j is the vehicle with the smallest positive virtual inter-distance to V_i among all the vehicles whose paths cross the one of V_i. If the vehicle finds that it has no leader, meaning that there are no other vehicles on its path, it can drive at its nominal velocity, without any risk of collision. Otherwise, it regulates its velocity in order to keep a safety inter-distance to its leader vehicle V_j. The same procedure is followed by all the other vehicles. This procedure, described in Algorithm 3.1, allows all the AD vehicles to cross the roundabout safely in a distributed manner.

Let us recall the hypotheses necessary for the proper operation of this approach. First, we suppose that the poses $[x,\ y,\ \theta]^T$ are well estimated (with negligible errors) for all the vehicles. We also assume that there is a perfect communication between all the vehicles without messages loss and negligible delay. Another assumption is

Algorithm 3.1 Elementary roundabout crossing procedure for AD vehicle V_i. In this case, N_i corresponds to the trajectory of AD vehicle V_i.

Require: $[x_i,\ y_i,\ \theta_i], N_i$ ▷ AD vehicle pose and trajectory
$d \leftarrow +\infty$
Send $(x_i,\ y_i,\ \theta_i,\ N_i)$ ▷ broadcast of the AD vehicle pose
$[V_1,\ V_2,\ \ldots,\ V_c] \leftarrow$ Receive() ▷ collect information about other communcating AD or MD vehicles
$s_{L_i} \leftarrow$ CurvAbscissa$([x_i,\ y_i,\ \theta_i],\ N_i)$
for $j = 1 : c$ **do**
    $n \leftarrow$ FindFirstCommonNode(N_i, N_j) ▷ compute trajectories intersection point (if it exists)
    if $n = \emptyset$ **then**
        Continue ▷ V_j does not cross the path of V_i
    else
        $s_{L_j} \leftarrow$ CurvAbscissa$([x_j,\ y_j,\ \theta_j],\ N_j)$
        $d_{ij|n} \leftarrow$ Interdistance$(s_{L_i},\ s_{L_j},\ n,\ N_i,\ N_j)$ ▷ compute virtual curvilinear inter-distance
        if $d_{ij|n} < 0$ **then**
            Continue ▷ V_j goes after V_i
        else
            $d \leftarrow \min(d, d_{ij|n})$
            $leader \leftarrow V_j$
        end if
    end if
end for
if $leader = \emptyset$ **then**
    Continue ▷ go with nominal velocity
else
    LongControl$(leader, d)$ ▷ platooning with leader
end if

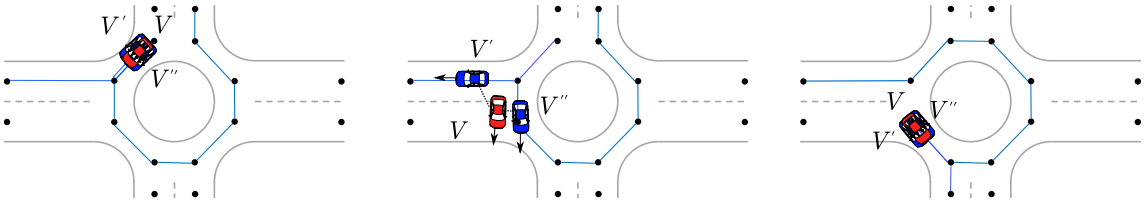


Figure 3.2: Mixed scenario. (a) Generation of the virtual vehicles $V^{'}$ and $V^{''}$as copies of vehicle V and their paths. (b) Branching scenario, the two vehicles $V^{'}$ and $V^{''}$follow their virtual paths until the most unlikely copy w.r.t. the real pose of V is deleted. (c) Re-computation of the virtual copies and their paths according to the new branch.

that all the paths are known and shared, which is reasonable for cooperative AD vehicles, but not for MD cars. For this more general use-case, a method to handle this issue will be presented hereafter.

3.2.3 Extension of the Method with Virtual Copies to Handle MD Vehicles

The approach explained previously shows how to handle scenarios in which only communicating AD vehicles are involved. In a more general context, one has to deal with the possible concomitant presence of both autonomous and MD vehicles. While it is reasonable to assume that AD vehicles can share their intended paths, it is not the case for MD vehicles, as they are not known a priori, since the route depends on the driver's intentions. Regarding the MD vehicles, we suppose only that their localization w.r.t. the map frame is known with low uncertainty and known by all the vehicles in the surroundings. This can be done by supposing either to have vehicles that share their pose through wireless communication or to have an intelligent infrastructure able to localize the vehicles in the environment and then broadcast the vehicles it perceives.

A simple way to extend the approach proposed previously is to make it handle manually driven cars by predicting their unknown paths. For this reason, a new approach that replaces a MD vehicle V by two copies $V^{'}$ and $V^{''}$ when a lane splits in two has been proposed. The main idea behind this is to assign to each virtual copy of V the two extreme paths w.r.t. the current position of V, that is the first exit branch for $V^{'}$ and the last exit branch (U-turn) for $V^{''}$, as it is shown in Figure 3.2a. When a branch occurs, the paths of the two virtual copies start to diverge from each other. Therefore, the map-matching algorithm will assign two different projections for each of them (see Figure 3.2b). Once the map-matching criterion becomes unlikely regarding to the current pose of V, the corresponding virtual copy is deleted and there is no longer ambiguities on the path of the manually driven car.

Then, two scenarios can occur:

1. The vehicle V is on an exit branch. In this case, the remaining copy keeps going on this branch until it disappears from the scenario. No new virtual copies need to be introduced.

2. The vehicle V is on an internal branch. In this case, the remaining copy is itself replaced if needed by two new virtual copies updated w.r.t. the new

configuration (Figure 3.2c).

As a matter of fact, it is fairly simple to implement this approach by simply replacing the line 3 of Algorithm 3.1, by the code of Algorithm 3.2 where the idea is just to manipulate the virtual copies in place of the MD vehicles, and the line 10 of Algorithm 3.1, with the code of Algorithm 3.3 to delete the irrelevant copy after a branching.

A threshold (used line 2 in Algorithm 3.3) defines the limit to which the vehicle can be far from the center of the lane. Such a threshold has been represented as a generic function f that depends both on $n_{V_j|L_j}$ and $\psi_{V_j|L_j}$. Such function gives the likelihood for a virtual vehicle to be matched on the different candidate lanes when an ambiguity arises. Moreocer, this approach works under the hypothesis that a MD vehicle always remains in the neighborhood of the lane (proximity constraint [40]). This guarantees that at least one of the virtual copies will remain after a branching.

Algorithm 3.2 Algorithms for virtual copies generation.

1: $[V_1,\ V_2,\ \ldots,\ V_c] \leftarrow$ receive()
2: **for** $j = 1 : c$ **do**
3: **if** isManual(V_j) **then**
4: $[N_j^{'},\ N_j^{''}] \leftarrow$ ExtremeTraj($[x_j,\ y_j,\ \theta_j]$)
5: $[V_j^{'},\ V_j^{''}] \leftarrow$ Copy(V)
6: $V_j^{'} \leftarrow$ UpdateTraj($N_j^{'}$)
7: $V_j^{''} \leftarrow$ UpdateTraj($N_j^{''}$)
8: Add($V_j^{'},\ V_j^{''}$)
9: Remove(V_j)
10: **end if**
11: **end for**

Algorithm 3.3 Algorithm for virtual copies deletion

1: $[s_{L_j}, n_{L_j}, \psi_{L_j}] \leftarrow$ CurvCoords($[x_j, y_j, \theta_j], N_j$)
2: **if** isVirtual(V_j) AND $f(n_{L_j},\ \psi_{L_j}) < Threshold$ **then**
3: Remove(V_j)
4: **end if**

3.2.4 Analysis and Verification of some Method Properties

In order to validate the algorithm presented before, we focus on verifying some properties that this method must satisfy in order to be considered a feasible approach. Let us first prove its correctness. In other words, the absence of unwanted behaviors during the execution of this method. In particular, the valodation of absence of "deadlock" (all the vehicles are waiting to cross the roundabout) and "starvation" (one vehicle is waiting indefinitely) will be investigated throughout this part. To do so, the problem has been decomposed in two possible situations. In the first one, the case in which it exists a total ordered sequence among the vehicles is considered, while in the second one, the dual situation is taken into account. Let us remind that every vehicle computes its own order sequence based on the current configuration of the other vehicles w.r.t. its own path (notice that in such a case vehicles can be real vehicles or virtual instances of them).

3.2.4.1 Totally Ordered Case

The first case is the situation where there exists a totally ordered sequence for the vehicles. This leads to a nominal virtual platooning scenario, where all the vehicles go through the roundabout one after the other, following the common order. It is quite easy to show that this occurs when, for any pair of paths, there is either no intersecting node or the first intersecting node is unique.

Let $V_1, V_2, \ldots, V_c$ be a set of c vehicles crossing a roundabout and $N_1, N_2, \ldots, N_c$ their respective paths. Supposing that for any pair of paths (N_i, N_j), either $N_i \sqcap N_j = \emptyset$ or $N_i \sqcap N_j = N_j \sqcap N_i = n$, where $N_i \sqcap N_j$ corresponds to the first common node between N_i and N_j relative to N_i, i.e., w.r.t. the ordered sequence of links of N_i.

A proof by contradiction has been used to demonstrate that there exists a total order between the vehicles. In this section, the subscript indexes have been added to a node in order to indicate the result of the operator $\sqcap$. In other words, the notation n_{ij} corresponds to the first common node between N_i and N_j. This is useful to distinguish cases where N_i and N_j intersect more than once, i.e. $N_i \sqcap N_j \neq N_j \sqcap N_i$.

Suppose there is no total order, that is there exists two vehicles V_i, V_j for which V_i decides that it goes before V_j:

$$d_{i,n_{ij}} < d_{j,n_{ij}} \quad \text{with} \quad n_{ij} = N_i \sqcap N_j, \tag{3.4}$$

and V_j also decides to go before V_i:

$$d_{i,n_{ji}} < d_{j,n_{ji}} \quad \text{with} \quad n_{ji} = N_j \sqcap N_i. \tag{3.5}$$

However, as it has been supposed that $N_i \sqcap N_j = N_j \sqcap N_i$, which leads to $n_{ij} = n_{ji}$.

The two relations (3.4) and (3.5) lead to a contradiction. This concludes the proof. This conclusion proves that in this case, if there is only one common intersection point between each couple of vehicles, a total order between vehicles exists.

Following this reasoning, one can say that, if a total order between the vehicles exists, it is not possible that a configuration where a vehicle cannot enter in the roundabout occurs.

In particular, considering the last vehicle V_k of a given totally ordered sequence of vehicles. If a new vehicle V_j arrives, this vehicle will be placed virtually behind vehicle V_k if the following conditions hold true:

- A total order between vehicle exists (i.e. all the vehicles intersect in the same point)
- The length of the distance from the beginning of the interaction zone and the intersection point is the same for every entering branch

The second condition is necessary for preventing situations where a vehicle coming from a branch that is far ahead w.r.t. some others will be blocked for an infinite time. This situation can happen because, in our approach, the total crossing order is decided considering only the relative inter-distance. In other words, a vehicle has to wait an infinite amount of time before entering into the roundabout if an infinite vehicles flow is coming from a branch that is much close to the common intersection point.

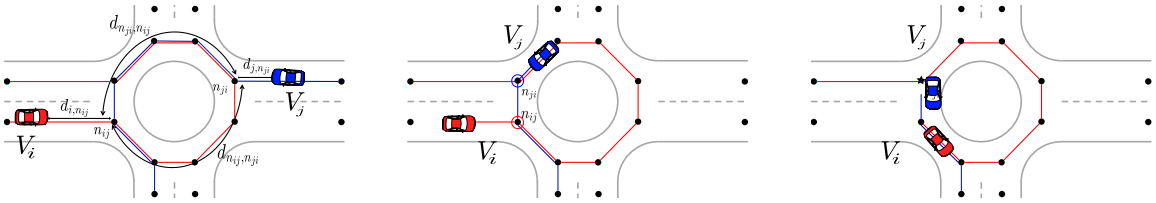


Figure 3.3: Dangerous cases: (a) Differently ordered case with the involved quantities. (b) Both vehicles are leaders at the same time, $N_i \sqcap N_j = n_{ij}$ for V_i and $N_j \sqcap N_i = n_{ji}$ for V_j. (c) V_i becomes leader of V_j as it overcomes n_{ji} and $N_i \sqcap N_j = n_{ij}$ for V_i and $N_j \sqcap N_i = n_{ij}$ for V_j.

3.2.4.2 Differently Ordered Case

We suppose now that the first intersecting node between two paths can be different, i.e., $\exists i, j$ such that $N_i \sqcap N_j = n_{ij}$ and $N_j \sqcap N_i = n_{ji}$ with $n_{ij} \neq n_{ji}$. The objective is now to show that no deadlock can occur even in this case.

Let us suppose a deadlock configuration in which both vehicles are waiting for the other to cross the roundabout. This means that

$$d_{i,n_{ij}} > d_{j,n_{ij}} \qquad \text{and} \qquad d_{j,n_{ji}} > d_{i,n_{ji}}. \tag{3.6}$$

We want to prove that this configuration cannot occur. Based on Figure 3.3a, one can write the following expressions:

$$d_{j,n_{ij}} = d_{j,n_{ji}} + d_{n_{ji},n_{ij}} \text{ and } d_{i,n_{ji}} = d_{i,n_{ij}} + d_{n_{ij},n_{ji}}, \tag{3.7}$$

where $d_{n_{ji},n_{ij}} > 0$ and $d_{n_{ij},n_{ji}} > 0$ are the curvilinear distances from the node n_{ji} to the node n_{ij} and from the node n_{ij} and n_{ji} according to the flow direction of the roundabout.

taking the relation $d_{i,n_{ij}} > d_{j,n_{ij}}$ and substituting the corresponding values, the following inequalities will be obtained:

$$\begin{aligned} d_{i,n_{ij}} > d_{j,n_{ij}} \quad &\Leftrightarrow \quad d_{i,n_{ji}} - d_{n_{ij},n_{ji}} > d_{j,n_{ji}} + d_{n_{ji},n_{ij}} \\ &\Leftrightarrow \quad d_{i,n_{ji}} > d_{j,n_{ji}} + d_{n_{ji},n_{ij}} + d_{n_{ij},n_{ji}} \\ &\Rightarrow \quad d_{i,n_{ji}} > d_{j,n_{ji}} \end{aligned} \tag{3.8}$$

This contradicts the deadlock condition (3.6). The same reasoning can be generalized to any number of vehicles.

Even if the absence of deadlock has been proved, there still exist some cases in which potentially unsafe configurations can arise.

Figure 3.3b shows a case in which, as it was said before, all the vehicles think to be the leaders, in fact we have $d_{i,n_{ij}} < d_{j,n_{ij}}$ and $d_{j,n_{ji}} < d_{i,n_{ji}}$. Such configuration shows clearly that both V_j and V_i think to be the leader of the (virtual) platoon and, for this reason they are free to move at their nominal speeds without taking care of the other car as an obstacle. As a consequence, nobody performs control w.r.t. the other, and both the vehicles proceed with their nominal velocities. This situation may lead to severe consequences. As one can see from figure 3.3c, if the virtual distance between n_{ij} and n_{ji} is not sufficiently large, we can have insufficient space to brake V_j when V_i overcomes the node n_{ij}. In facts, once V_i overcomes the node n_{ij}, V_j becomes the follower of V_i. In such situation, if V_j is too close to n_{ij} a collision can occur (Figure 3.3c).

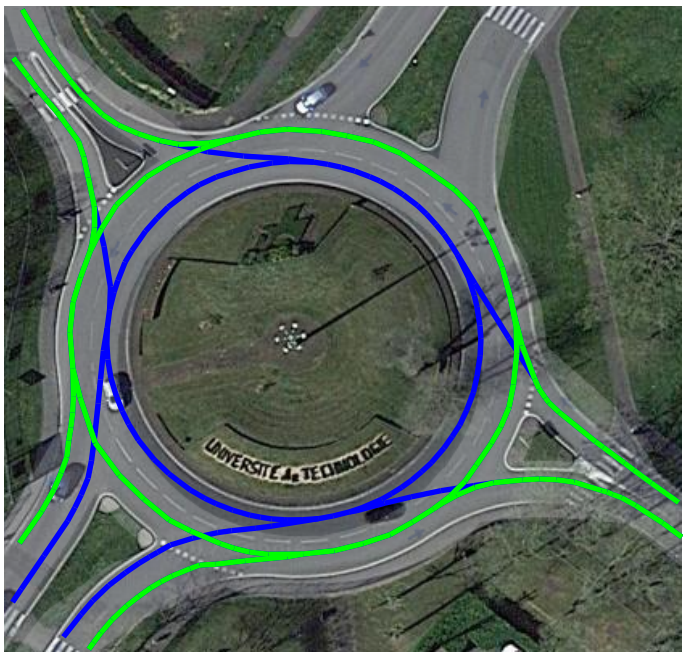


Figure 3.4: The "Guy Deniélou" roundabout in the city of Compiègne, France. The green line represents the outer lane of the roundabout while the blue one is the inner one. The north-east exit of the roundabout was not mapped at the time we conducted this study and was ignored in this study.

3.2.5 Simulation Study

To study the behavior of the algorithm, different experimental scenarios have been considered. Simulations have been carried out considering the map representation of a large roundabout in the city of Compiègne (see Figure 3.4). This roundabout presents two lanes into the central part and two-lanes roads for some branches. In the simulations, only the outer lane of the roundabout and the right side lane for the branches with multiple lanes have been considered. This configuration is represented as the green lines in Figure 3.4. The generalization to multiple lanes will be describebed later in this chapter. Remember that, in this context, it has been supposed to have a perfectly reliable V2V communication between MD and AD vehicles.

In order to simulate the MD and AD vehicles motion, it has been decided to constrain their motion along the polylines of the HD map. This allows to focus the attention only on the longitudinal component of the motion, which is the most relevant for preventing collisions between vehicles in our formalism.

One other reason of this choice is that, in absence of a received pose to map-match on the HD map, this approach allows also to simulate the map-matching of a real MD vehicle on the map polylines, without knowing its Cartesian pose. To do so, the analytical the evolution of each MD and AD vehicle curvilinear abscissa along the HD map polylines has been computed, according to a given dynamic evolution model (in curvilinear coordinates) and the previously seen formalism.

In order to control the vehicles during platooning, a longitudinal control law has been used. This control law is based on the inter-distance between the leader vehicle (virtual or real) and the follower which implements the law. In a simple case, the desired value of the curvilinear inter-distance is a linear function of the velocity v of the following vehicle in the form:

$$d = d_0 + h \cdot v, \tag{3.9}$$

where d_0 is the standstill distance and h is the time headway.
To do the feedback, a proportional-derivative control law has been used. That law computes the acceleration of the follower as below:

$$a(t) = \alpha_1 \left(s_l(t) - d - s_f(t) \right) + \alpha_2 \left(\dot{s}_l(t) - \dot{s}_f(t) \right), \tag{3.10}$$

where α_1 and α_2 are two real valued gains, $s_l(t)$ and $s_f(t)$ the curvilinear abscissa of the leader and the follower vehicles respectively, and $\dot{s}_l(t)$, $\dot{s}_f(t)$ their corresponding velocities. Saturation functions in the form $v(t) = \max(v_{max}, v(t))$ and $v(t) = \min(v_{min}, v(t))$ are considered to bound the velocity into fixed limits. Acceleration saturation has also been added. As this study is done in simulation, the control law calculates the acceleration and not the torque (acceleration or braking) as it does with a real car.

3.2.5.1 Safety Diagram

Let us introduce a concept called "Safety diagram" in order to evaluate the behaviors of the vehicles in a given scenario [66]. The aim of this diagram is to show when safety margins are violated, i.e., when the curvilinear inter-distance w.r.t. the first non-virtual obstacle along a vehicle path is too short given its current velocity.

Consider a diagram as the one in Figure 3.5, where the inter-distance is plotted as a function of the velocity of the vehicle. One can see that Equation (3.9) draws a boundary between the safe and unsafe zones. If a point (v, d) is above that line, it means that the inter-distance d is sufficiently large for the velocity v. Inversely, if the point (v, d) is found under the boundary, it means that d is too short for the vehicle velocity v, i.e., an unsafe case occurs. Finally, if (v, d) lies on the boundary, it means that the vehicle is perfectly at the desired curvilinear inter-distance in the platoon w.r.t. its velocity v.

To illustrate a typical scenario that occurs in our algorithm with the corresponding safety diagram interpretation, let us consider a leader vehicle driving at a constant velocity v_l and a follower in virtual platoon w.r.t. the leader. Figure 3.5 shows the behavior of the follower that is entering the roundabout at time $t = t_0$ and finds a leader vehicle ahead of him. From t_0 to t_1 the follower accelerates to reach the desired inter-distance from its leader until the speed saturation occurs at t_1. Then from t_1 to t_2, it continues to move at the maximum allowed speed v_{max}, decreasing the inter-distance until t_2 when it starts braking as it catches the leader. Finally, at t_3, the follower reaches the optimal inter-distance w.r.t. the leader and continues driving at the same velocity as the leader.

3.2.5.2 Monte Carlo Simulation

To test our approach, we have developed a simulator able to generate random vehicles over time, with stochastic trajectories. The simulator implements the algorithm described in section 3.2, allowing to choose the frequency for generating both autonomous and manually driven vehicles, in order to cover every possible scenario.

Figure 3.6 shows the resulting safety diagram for a given simulation with a dozen vehicles mixing both autonomous and manually driven ones. One can see that all the trajectories tend to converge towards the safety boundary which is the optimal inter-distance. The control law may allow the vehicles to overshoot the optimal inter-distance, but as one can see, it does not deviate significantly under the safety boundary.

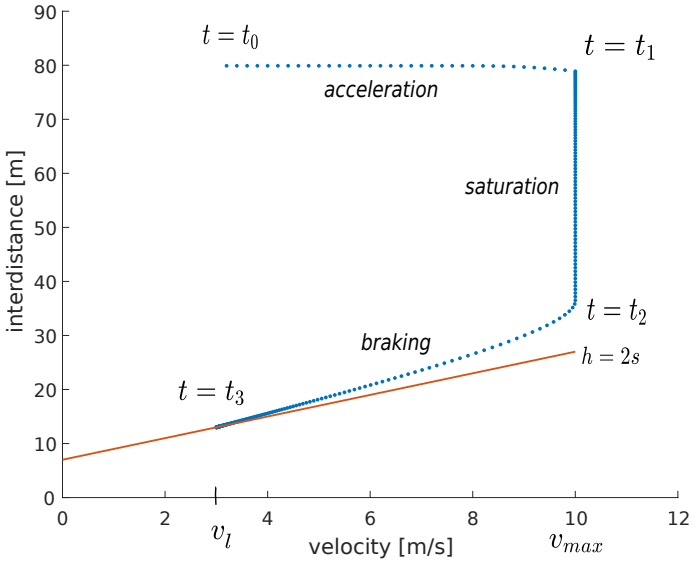


Figure 3.5: Safety diagram illustration. The blue dots represent the (v, d) values of the following vehicles w.r.t. the leader with velocity $v_l = 3m/s$ and $v_{max} = 10m/s$. The orange line represents the safety boundary with $d_0 = 7m$ and $h = 2s$.

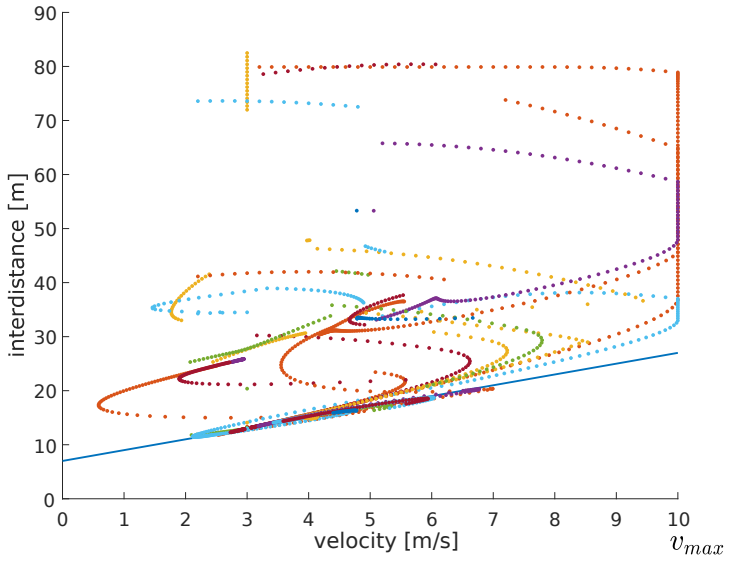
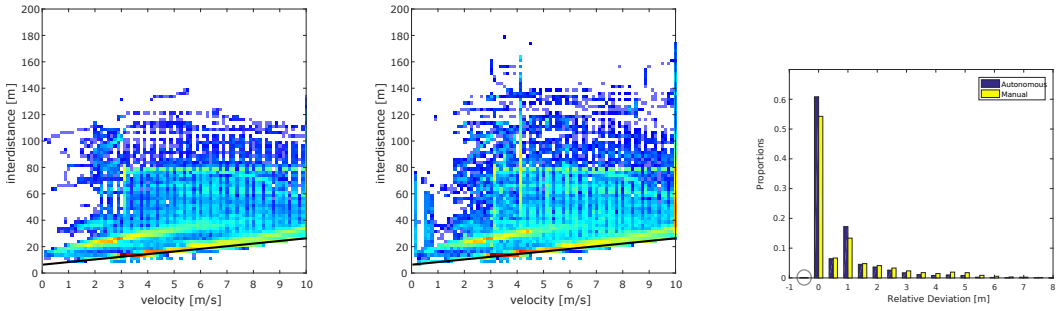


Figure 3.6: Safety diagram for a stochastic simulation with a dozen vehicles.



(a) AD vehicles only. (b) AD and MD vehicles. (c)

Figure 3.7: (a-b) Empiric joint log-probability density function after a Monte Carlo simulation with 100 iterations. (c) Distribution of the relative deviation between the actual and optimal inter-distance. The circle depicts the unsafe region with a proportion of about 1%.

In order to present an exhaustive analysis of this algorithm, a Monte Carlo simulation has been implemented. The aim is to extend the analysis to cover as many cases as possible, in order to study statistically the performances of the algorithm. results of two Monte Carlo simulations have been reported in the following figure:

1. AD vehicles only; (Figure 3.7a)
2. AD and MD vehicles. (Figure 3.7b)

In Figures 3.7a and 3.7b, the the empiric joint probability density function of the velocity and the corresponding inter-distance using a logarithmic scale is displayed. One can see that most of the points are located above the safety boundary. However, there are also few points in the unsafe region. Several reasons can explain this behavior. First, the imperfect tuning of the controller can lead the vehicles to have some overshoots over the optimal boundary. Second, the differently ordered case presented in section 3.2.4.2 also allows vehicles to have their inter-distances lower than the optimal ones. Figure 3.7c shows the distribution of the relative deviations from the optimal inter-distance computed as follows:

$$e_r = \frac{d_v - (d_0 + hv)}{d_0 + hv} = \frac{d_v}{d_0 + hv} - 1, \tag{3.11}$$

where d_v is the actual distance w.r.t. the first non-virtual obstacle. Moreover, the proportion of points that deviate more than the 5% into the unsafe region has been computed and, it has been shown that this percentage is in the order of 1%.

From these figures, one can see that the algorithm performed similarly in both cases. The autonomous only scenario allows the vehicles to drive with inter-distances closer to the optimal ones (which optimizes traffic flow).

3.2.6 Discussion

The results reported in Section 3.2.5 highlight the efficiency of our proposed approach to solve the roundabout crossing problem for an AD vehicle. However, in spite of the quality of the results, we need to point out some weaknesses and limitations of the presented method.

First, the proposed formalism does not consider the existence of any priority constraint between vehicles. Priorities and, as a consequence, the right of way, are

aspect that naturally arise during AD vehicles navigation in mixed traffic. For this reason, they cannot be disregarded during the design of a planning strategy. In this particular use-case, it exists a priority constraint between vehicles in the roundabout ring and vehicles that are coming from the roundabout entering branches.

Second, in the navigation strategy, the computation of the crossing order between the different vehicles is based only on the comparison of relative inter-distances. This method does not consider the relative speed between the AD vehicle and the other MD one to be included in the crossing order computation. Such a method may lead to poor performance when the relative speed difference is significant. Furthermore, it does not allow to identify when a vehicle is stopped (for example at a give way sign before entering into the roundabout), leading again the algorithm to poor performance.

Finally, the previous approach supposes that vehicles are represented as dimensionless points moving along the HD map polylines, disregarding the uncertainty on the object position estimation and the size of the objects.

For these reasons, we propose, in the following part of the chapter, to incorporate in the previously presented formalism the main aspects that have been pointed out in this section. The main idea is to propose a new version of the roundabout insertion algorithm that is more realistic and more scalable to a real life scenario.

3.3 Single-lane Roundabout Crossing with priority and interval occupancy

3.3.1 Introduction

In the previous section, it has been shown that using virtual vehicles along the lanes of a HD map is very efficient to control the longitudinal behavior of an AD vehicle when crossing a roundabout.

Then, an extension of this method to handle cases with both AD and MD vehicles is presented. However, some constraining hypothesis have been made about the MD vehicles behavior. In particular, we assume that the localization of every MD vehicle is known with low uncertainty and that MD vehicles drive with a straightforward behaviors i.e. without breaking traffic rules, which is not always the case.

In addition, it has been also considered that direct communication exists between both AD and MD vehicles. This means that they are able to exchange information about their states respectively. Unfortunately, nowadays this assumption is of course unrealistic, because there are only a few MD vehicles on public traffic equipped with this kind of systems.

As a consequence, during autonomous navigation, AD vehicles have to estimate thanks to their own on-board perception system the state of surrounding MD vehicles.

Finally, in the previously described approach, traffic rules in roundabout navigation have not been explicitly taken into account. In particular, the priority constraint between vehicles on the roundabout ring and incoming vehicles on entering branches has been disregarded.

In this section, we propose an extension of the previously discussed strategy to incorporate gradually the aforementioned conditions in order to provide a more general navigation algorithm to be exploited in real MD traffic driving scenarios.

Then, an extended experimental evaluation will be provided to show the feasibility

of this approach on a real-traffic scenario and to focus on the resolution of some issues that arise when virtual vehicle techniques are applied to roundabouts. The main questions of this part are listed hereafter:

- A safe, priority-preserving and not overly cautious decision method for one-lane roundabout crossing for an AD vehicle among a MD vehicles flow;
- An extension of the curvilinear coordinates virtual platooning to an interval-based curvilinear formalism to include both vehicles sizes and uncertainties in vehicles positions in the navigation algorithm;
- The use of the virtual vehicle concept to handle prediction of unknown intention of vehicles and to handle the application of virtual vehicle methods to roundabout cycles;
- A practical evaluation of the aforementioned algorithm in terms of safety and fluidity of the insertion with realistic simulations and real experiments;

The rest of this part is organized as follows. In the next section, an introduction a curvilinear formalism with intervals along with HD maps is presented. Then, in section 3.3.3, we come back on the concept of virtual instances and its application to roundabouts use cases. In section 3.3.4, a discussion about the main principles and rules that we use in our approach is carried out, while in section 3.3.5, it will be explained how the priority constraints and the right of way have been taken into account. Finally, an explaination of the insertion maneuver for roundabout is provided in the cases where MD vehicles are involved.

3.3.2 Interval-based Curvilinear Inter-distances

Let us present an interval-based formalism to compute curvilinear inter-distances between several vehicles. Such a concept can be seen as a generalization of point-based inter-distances that we used in the previous algorithm (3.2). The main idea is to represent objects positioning w.r.t. the HD map with an interval rather than a point on the polyline. Such a formalism is useful to encompass the positioning uncertainty and the size of the MD vehicles. For this, the HD map formalism and curvilinear coordinates computation are again exploited to develop a new formalism.

Based on the curvilinear formalism introduced in section 3.2.1, let us generalize the virtual curvilinear inter-distance of Equation (3.3) to the case of interval curvilinear abscissa. For an interval $[\underline{s}_i, \overline{s}_i]$, let us define $\left[\underline{s}_{i|n}, \overline{s}_{i|n}\right]$ the lower and upper curvilinear abscissa of the vehicle w.r.t. a given node n using Equation (3.2). As said previously, these quantities are negative and increase when the vehicle approaches the node n. Defining the interval-based virtual curvilinear signed inter-distance $d^*_{i,j|n}$ between V_i and V_j, whose paths intersect at a node n as the signed distance between the front of V_j and the back of V_i we obtain:

$$d^*_{i,j|n} = \overline{s}_{j|n} - \underline{s}_{i|n}. \tag{3.12}$$

Therefore, $d^*_{i,j|n} < 0$ means that w.r.t. node n, V_i is virtually ahead of V_j, i.e., they are in the first configuration illustrated in Figure 3.8c. It is important to note that $d^*_{i,j|n} > 0$ does not mean that V_j is ahead of V_i, indeed, any of the other configurations in Figure 3.8c would lead to this case. Contrary to the case where a

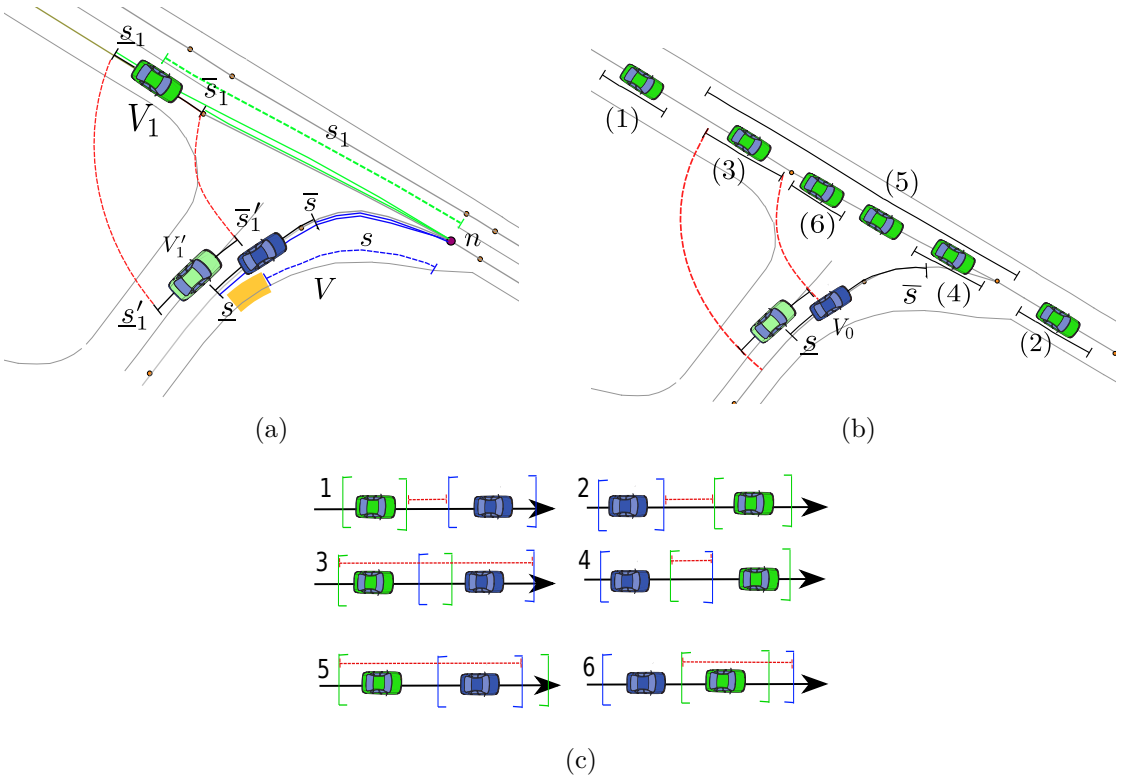


Figure 3.8: (a) Classical virtual platooning (dashed) and its extension to intervals (solid). V is the ego-vehicle and $V_1^{'}$ is a virtual instance of V_1. (b) Situation with 6 other vehicles (all the possible relative locations) Occupancy is projected onto the road map. (c) Intervals overlapping.

single curvilinear abscissa is used, we have $d^*_{i,j|n} \neq -d^*_{j,i|n}$, i.e., this Equation is not symmetric. To better understand this concept, Figures 7.7a, 7.7b and 3.8c explain such concept in the case of a simple T intersection. In the rest of the paper, for simplicity, $d^*_{i,j|n}$ will be denoted $d^*_{i,j}$ when there is no ambiguity.

3.3.3 Unknown MD Vehicles Intentions in Graph Cycles

In order to encompass all the possible behaviors of a MD vehicle, virtual instances can be used to represent all the behaviors. The concepts that have been introduced in section 3.2.3 are again exploited to provide a method that can help to solve some situations of relative positioning inside the roundabout ring that arise in cyclic graphs. Considering Figure 3.9 and applying the virtual platooning algorithm as in section 3.2, one can see that the paths of V_0 and the instance V_1 of the truck have n_i as the first intersecting node (from the point of view of the AD vehicle V_0). This means that V_1 is virtually behind V_0 even if it is clearly not the case because the truck can be seen both behind and ahead V_0, depending on the point of view. Moreover, if one considers long objects, as for example a truck with a trail, this ambiguity is more flagrant w.r.t. cases where objects with small size are involved. This particular behavior is due to the circular shape of the roundabout ring and in some cases it may produce an erroneous representation of the scenario. In fact, V_0 needs to consider the presence of the truck during the insertion maneuver as a vehicle to follow, and it also needs to consider its presence as a possible incoming vehicle on the left side of the roundabout. To overcome this issue, let us consider again the instance $V_1^{'}$ that represents the same vehicle but with a different path. One

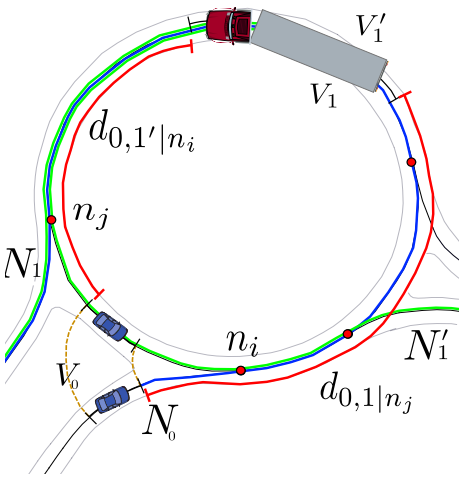


Figure 3.9: The truck trajectories are estimated considering several virtual instances of the same vehicle according to section 3.2 assigning at each instance a possible trajectory. In this case, the truck can be both ahead and behind the AD vehicle.

can see that it has n_j as the first intersecting node with the path of V_0. This means that, in this case, V_0 is behind the instance $V_1^{'}$. This representation is the dual of the previous one, where we claimed that the instance V_1 of the truck was virtually behind V_0. As a consequence, in a roundabout it is possible to have several instances of the same object with a different relative positioning w.r.t. the AD vehicle. In other words, an MD vehicle can be both virtually ahead and behind the AD vehicle. This method enables the AD vehicle to overcome the problem of the roundabout loop and allows it to consider all the possible MD vehicles configurations in the scenario.

3.3.4 Insertion Strategy

Let us assume that the state of an MD vehicle V_i is represented as

$$V_i = [\underline{s}_i, \overline{s}_i, v_i, N_i]\,, \tag{3.13}$$

where $[\underline{s}_i, \overline{s}_i]$ are the lower and upper bounds over the curvilinear occupancy of V_i encompassing both the size and the uncertainty bounds over its position estimate, v_i is its estimated longitudinal speed and P_i its predicted path.

The values of $\underline{s}_i$ and $\overline{s}_i$ are computed as explained in section 3.3.2. From the perspective of the ego AD vehicle, the state of a nearby vehicle is typically provided by a perception system able to detect, track and map-match.

Moreover, it has been decided to add the MD velocities v_i to the state because it allows to better take into account the dynamic behavior of the MD vehicles. Adding information about MD vehicles speeds allows to better encompass both MD vehicles behavior and to implement a more complex roundabout insertion strategy that also handles priority constraints between vehicles. This will be detailed in section 3.3.6.

In this work, in order to be compatible as much as possible with most of the state-of-the-art road user detection algorithms, no other assumption has been made on the information about surrounding vehicles. In particular, the reader may refer to [13], where it is explained how the quantities in Equation (3.13) are provided to the system.

Here the AD vehicle is constrained to navigate only on the outermost lane of the roundabout. In other words, it is not allowed for the AD vehicle to overtake and

change lane during the roundabout crossing. With this simplification, the navigation algorithm only needs to control the longitudinal motion of the AD vehicle to perform the task, the lateral control being done by path following.

Moreover, to cross a roundabout successfully, one needs to take into account not only the safety inter-distance w.r.t. the vehicle ahead, but also the priority relationships in the roundabout scenario. In a roundabout, the priority lanes are situated inside the roundabout ring while the non-priority ones are in the entering branches. Bearing in mind such concepts, three rules to describe the ideal behavior of an AD vehicle should have have been inferred.

An AD vehicle:

R1) has to keep a safe inter-distance w.r.t. the vehicle ahead;

R2) has to respect the traffic rules (vehicles inside the roundabout have the right of way);

R3) should avoid stopping on the carriageway as much as possible.

This means that a vehicle on a non-priority lane is allowed to enter into the roundabout only if the insertion maneuver does not influence the behavior of another vehicle with a higher priority rank. In other words, the entering vehicle is not allowed to force a priority vehicle to decrease its speed. A priority vehicle follows its reference speed profile and performs an inter-distance regulation only with respect to vehicles that have the same priority level. It is also desired that the AD vehicle makes an insertion as smooth as possible without making a stop at the entrance of the roundabout which requires it to anticipate the behavior of the other vehicles.

3.3.5 Roundabout Lanes Classification and Priorities

In accordance to the problem statement and the three rules listed in section 3.3.4, we propose to decompose a roundabout into three types of zones as illustrated in Figure 3.10. Each zone describes sub-steps of the insertion maneuver and a different priority rank, as follows:

a) The *Decision Zone* (in green in Figure 3.10) is before the merging into the roundabout ring. In this zone, the AD vehicle does not have priority w.r.t. vehicles in the roundabout. It has to evaluate the possibility of a safe insertion in the roundabout without violating priority constraints.

b) The *Transition Zone* (in yellow in Figure 3.10) is the last part of the entering lane where it merges with the roundabout ring. In this part, the AD vehicle performs a transition to enter into the roundabout. When the AD vehicle is in that zone, a safety inter-distance w.r.t. a potential incoming MD vehicle on the roundabout ring must be kept in order to allow a safe insertion.

c) The *Ring Zone* (in red in Figure 3.10) corresponds to the roundabout ring. In this zone, the insertion maneuver is completed and the AD vehicle follows the nearest MD vehicle in the roundabout or drives at its nominal speed if it is alone.

d) The *Exit Zone* (in dark gray in Figure 3.10) is the zone where the AD vehicle leaves the roundabout and continues its navigation following its path.

As a consequence, the crossing of the whole transition zone must be taken into account in the decision-making procedure. Once a vehicle enters into that zone, it can no longer change its decision. If it needs to stop, it means that the decision of entering into the roundabout was wrongly taken which makes the approaching vehicle decelerate in order to avoid a collision. Finally, once the AD vehicle has

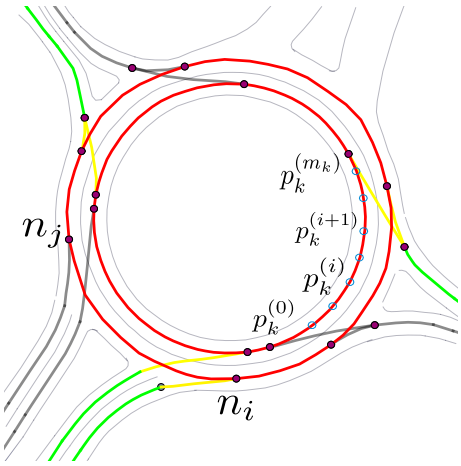


Figure 3.10: A roundabout with its HD map representation. The decision zones are green, the transition zones are yellow and the ring zone is red. The roundabout exits are blue. For a given link, the nodes and shape points are shown.

crossed the transition zone, it gains the same priority as all the vehicles in the roundabout ring.

3.3.6 Intervals-based Virtual Platooning

A well-known technique to achieve intersection crossing is to establish a crossing order between incoming vehicles [69]. However, when using intervals to represent the curvilinear occupancy of vehicles, cases where no total order between vehicles may arise if one applies virtual vehicle methods as in [69] and section 3.2. This is due to possible intervals overlapping.

Let us consider the case illustrated in Figure 7.7a, where the trajectory of the AD vehicle (in blue) crosses the one of a MD vehicle (in green) at a point n. Let us define $[\underline{s}, \overline{s}]$ the curvilinear occupancy of the AD vehicle w.r.t. the origin n, and similarly $[\underline{s}_i, \overline{s}_i]$ for a MD vehicle V_i. Figure 3.8c illustrates the six possible relative positions between the AD vehicle and the MD ones with their corresponding virtual projections.

Note that if there is no priority constraint between the vehicles, using intervals does not lead to a unique order among them. Moreover, if there is no total order between the vehicles and no priority constraints, some deadlock situations may arise, as we previously discussed in section 3.2.4.

To overcome this issue, one needs to choose an insertion policy [68]. This issue is out of the scope of this work, as it has been chosen to consider that vehicles inside the roundabout have the highest priority. In Figure 3.8c, one can see that in all the cases (2), (3),..., (6), the AD vehicle cannot be guaranteed to be in front of the incoming MD vehicle. These cases should not occur when the AD vehicle goes through the transition zone otherwise it will be in contradiction with rule R2. If the AD vehicle cannot guarantee that such cases will not occur, the AD vehicle decreases its speed to let the incoming vehicle go ahead. Such maneuver can lead either to a safe stop at the give-way line or to a speed adaptation depending on the relative intervals overlapping over time. In other words, the AD vehicle tries to adapt as much as possible its behavior to let the other car go first and if it cannot, it performs a safe stop. Notice that the only case where the AD vehicle may expect to be in front of the MD vehicle is the case (1).

Let us consider the scenario depicted in Figure 3.9, where the AD vehicle V_0 is in

the decision zone of the roundabout and an incoming priority MD vehicle V_1 is in the ring zone. This scenario falls clearly into case (1), with $d^*_{0,1} > 0$.

Let us define t_0 as the time when the front part of the AD vehicle enters into the transition zone. At a given time $t < t_0$ (*i.e.,* when the AD vehicle was still in the decision zone), one can see that having

$$d^*_{0,1}(t_0) > d_{\text{safe}}, \tag{3.14}$$

is not sufficient for the AD vehicle to ensure a safe and priority preserving insertion maneuver according to rule R2. Indeed, if the speed v_1 of vehicle V_1 is greater than the speed v_0 of V_0, the inter-distance between the two vehicles will shrink over time. This shows that vehicle kinematics must be taken into account at the decision-making level.

Considering again rule R2, the AD vehicle needs to guarantee that Equation (3.14) will be satisfied during the whole insertion maneuver, *i.e.,* from the moment that the upper bound $\overline{s}$ enters the transition zone until the lower bound $\underline{s}$ leaves it. Let Δt be the time needed by the AD vehicle to go completely through the transition zone, *i.e.,* the back of the AD vehicle has exited it. The decision to enter the roundabout is taken if:

$$\forall t \in [t_0, t_0 + \Delta t], \quad d^*_{0,1}(t) > d_{\text{safe}}. \tag{3.15}$$

In order to guarantee the inequality of Equation (3.15), one needs to know how both $\underline{s}_{0|n}$ and $\overline{s}_{1|n}$ evolve over time. In this work, it has been assumed that vehicles drive at an almost constant speed. This assumption may seem simplistic, but it is representative of the driver behavior as indicated in the INTERACTION dataset. Indeed, within the roundabout ring of the INTERACTION dataset, the speed profiles of the MD vehicles have a standard deviation less than 1 m/s in average. Under this assumption, it is possble to use the formula $\Delta t = l/v_0$, where l is the length of the transition zone, and the kinematics of each interval can be expressed as follows:

$$\underline{s}_{0|n}(t) = \underline{s}_{0|n}(t_0) - v_0 \cdot (t - t_0), \tag{3.16}$$

$$\overline{s}_{1|n}(t) = \overline{s}_{1|n}(t_0) - v_1 \cdot (t - t_0). \tag{3.17}$$

Substituting Equation (3.16) and (3.17) in (3.15), we obtain

$$\underbrace{\overline{s}_{1|n}(t_0) - \underline{s}_{0|n}(t_0)}_{=d^*_{0,1}(t_0)} + (v_0 - v_1) \cdot (t - t_0) \geq d_{\text{safe}}, \tag{3.18}$$

The inequality (3.18) needs to hold $\forall t \in [t_0, t_0 + \Delta t]$.

If $v_0 > v_1$, it leads to

$$d^*_{0,1}(t_0) \geq d_{\text{safe}}. \tag{3.19}$$

It means that if the AD vehicle drives faster than V_1, it can insert if it is sufficiently ahead of V_1 at t_0.

Otherwise, if $v_0 < v_1$, the following case occurs:

$$d^*_{0,1}(t_0) \geq d_{\text{safe}} + \left(\frac{v_1}{v_0} - 1\right) l. \tag{3.20}$$

One can see that the relative speed v_1/v_0 needs to be taken into account in the decision and that if Equation (3.20) holds at t_0, it also holds for all the interval $[t_0,\, t_0 + \Delta t]$. This is particularly useful in the case where the AD vehicle accelerates from a low speed ($v_0 = 0$ in the case of a stop at give way) to enter into the

roundabout because it already encompasses all the speed changes of the AD vehicle in the decision. In other words, the decision taken at t_0 cannot change if the speed of the AD vehicle increases.

Inequality (3.20) allows the AD vehicle to decide if it has enough space to keep a safety inter-distance w.r.t. an eventual incoming vehicle, knowing its velocity and its occupancy at time t_0. In the case where Equation (3.20) is not satisfied, the ego vehicle slows down to perform a safe stop at the end of the decision zone (that coincides with the give way marking).

Nevertheless, once the speed of the AD vehicle is close to zero, it is difficult for the AD vehicle to find a sufficiently large gap to perform the insertion. This is due to the singularity present in Equation (3.20) when $v_0 = 0$. In fact, considering the function $h(v_0) = \left(\frac{v_1}{v_0} - 1\right)$, one can see that it has a peak towards $+\infty$ for $v_0 \to 0$. This degrades the performance of the algorithm once the AD has stopped: it will only insert once no incoming vehicle is present. To overcome this, it has been proposed to replace $h(v_0)$ with another function for the case $v_0 < v_1$. In particular, we look for a function that meets the following criteria:

1. As $v_0 \to 0$, the value of the function becomes less dependent on v_0.
2. For $v_0 = 0$ the value of the function depends at least on v_1.

The main idea is to have a function that allows to set a safety gap that depends at least only on the other vehicle speed v_1. This solution is convenient when the dynamic of the system is not well known and we need to perform a prediction without being too pessimist. In this work, a function with the following characteristics has been chosen:

$$\widehat{h}(v_0) = A\left(\frac{1}{2} - \frac{1}{1 + e^{-\alpha(v_0 - v_1)}}\right) \tag{3.21}$$

Where the two parameters A and α have to be tuned experimentally to make a good insertion maneuver. Equation (3.20) now becomes

$$\begin{cases} d^*_{0,1}(t_0) \geq d_{\text{safe}} & \text{if } v_0 > v_1, \\ d^*_{0,1}(t_0) \geq d_{\text{safe}} + \widehat{h}(v_0)l & \text{else.} \end{cases} \tag{3.22}$$

or equivalently

$$\begin{cases} d^*_{0,1}(t_0) - d_{\text{safe}} \geq 0 & \text{if } v_0 > v_1, \\ d^*_{0,1}(t_0) - d_{\text{safe}} - \widehat{h}(v_0)l \geq 0 & \text{else.} \end{cases} \tag{3.23}$$

Figure 3.11 illustrates the function $\widehat{h}$ for several values of A and α. As one can see, the singularity present in h for $v_0 = 0$ is avoided. For instance, for $\alpha = 1$ and $A = 10$, if the AD vehicle is at $v_0 = 0$, then it decides to enter if the inter-distance is around 50 m. To describe how to apply the aforementioned concepts in the case of a traffic flow, algorithm 3.4 details the insertion strategy in a general case. In such a case, one can notice that when the AD vehicle has at least one vehicle behind and one ahead, the prediction Equation 3.23 is done considering the vehicle ahead speed instead of the AD one. This allows to take into account the presence of an eventual vehicle ahead during the insertion maneuver. On the other hand, when there is at least one vehicle behind the AD car that does not satisfy Equation 3.23, the AD vehicle performs a safe stop at the give way, and it chooses the vehicle to follow for entering into the roundabout as the farthest object that does not satisfy Equation 3.23 in order to avoid unsafe configurations.

Algorithm 3.4 Roundabout insertion for an AD vehicle V_0.

Require: V_0, t_0

$[V_1,\ V_2,\ \ldots,\ V_c] \leftarrow$ Perception() ▷ Other road users
$\Delta t \leftarrow l/v_0$
$leader \leftarrow \emptyset, \quad d_{leader} \leftarrow -\infty$
$V_{RISK} \leftarrow []$ ▷ List of vehicles that present a risk
for $j = 1 : c$ **do**
    $n \leftarrow$ FindFirstCommonNode(N_0, N_j)
    if $n = \emptyset$ **then**
        Continue ▷ V_j does not cross the path of V_0
    else
        $d^*_{0,j|n}(t_0) \leftarrow \overline{s}_{j|n}(t_0) - \underline{s}_{0|n}(t_0)$
        if $d^*_{0,j|n}(t_0) < 0$ **then** ▷ Vehicle V_j in front of V_0
            if $d^*_{0,j|n}(t_0) > d_{leader}$ **then**
                $d_{leader} \leftarrow d^*_{0,j|n}(t_0)$
                $leader \leftarrow V_j$
            else
                Continue
            end if
        else ▷ Vehicle V_j behind or intersecting V_0
            if $d^*_{0,j}(t_0) \geq d_{\text{safe}} + \left(\frac{v_j}{v_0} - 1\right) l$ **then**
                Continue ▷ Contraint satisfied
            else
                $V_{RISK} \leftarrow V_{RISK} \cup [V_j, d^*_{0,j}(t_0)]$
            end if
        end if
    end if
end for
if $leader = \emptyset$ **then**
    $v_d \leftarrow v_n$ ▷ Go with nominal speed
else
    $v_d \leftarrow leader.speed$ ▷ Speed of vehicle ahead
end if
if $V_{RISK} \neq []$ **then** ▷ Change leader vehicle
    $leader \leftarrow \max_{d^*_{0,j|n}} [V_{RISK}]$
    $v_d \leftarrow leader.speed$
end if
SetSpeed(v_d) ▷ Perform longitudinal control

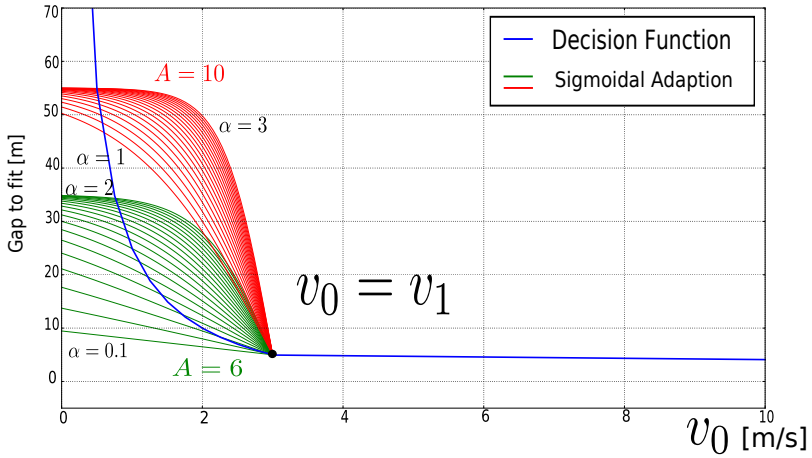


Figure 3.11: The behavior of the decision function with h and $\widehat{h}$ (Equation (3.22)) in terms of required inter-distance for several values of A and α for a fixed value of v_1, l and d_{safe}.

3.3.7 Simulation Results

To evaluate this approach, two simulators have been used. The complete description of both of them can be found in Chapter 2. The SUMO simulator [51] was used for microscopic traffic flow generation while a ROS-based simulation framework was used to implement the navigation algorithm and the AD vehicle dynamics. The coupling and synchronization of both simulators have been achieved with the TraCi SUMO library and its Python API. A detailed explanation on time synchronization and coupling of the two simulators can be found in the previous chapter and in [36].

Furthermore, we have imported in the SUMO simulation environment the HD map representation of the test-bed roundabout. This has been achieved with the Netedit and Netconvert tools included in the SUMO suite [48]. Figure 3.12 pictures an overview of the roundabout scenario in both simulators.

For each simulation, a random high density vehicle flow that meets the $\Delta TTIC_{min}$ criterion has been generated over a fixed time horizon $T = 200$ s. Each simulation contained between 200 and 400 seconds for a total amount of more than 1 hour of simulation. The number of vehicles for each flow randomly varied between 50 and 175, for a total amount of more than $5,000$ vehicles.

These limits have been chosen to capture a wide range of scenarios, starting from a sparse traffic flow until a denser vehicle stream. Moreover, to simulate both localization (AD vehicle) and perception (MD vehicles) uncertainty, we consider a ± 1 m bound to add to the curvilinear interval $[\underline{s}_i, \overline{s}_i]$ and $[\underline{s}, \overline{s}]$, which represents the projection of the vehicle footprint on its lane.

To experimentally validate this approach, several scenarios have been considered to generate high density traffic flows. Simulations have been carried out with the HD map representation of the roundabout displayed on Figure 2.1.

To quantify the performance, the distributions of the inter-distances w.r.t. the vehicle ahead and behind during the crossing of the transition zone have been computed. Figure 3.13 shows the inter-distance distributions for both the ahead and behind gaps. As one can see, the behind safety gap always meets the safety criterion. Conversely, considering the gap w.r.t. vehicles ahead, there is a slight violation of the safety bound. This violation is due to some controller imperfection and can be neglected.

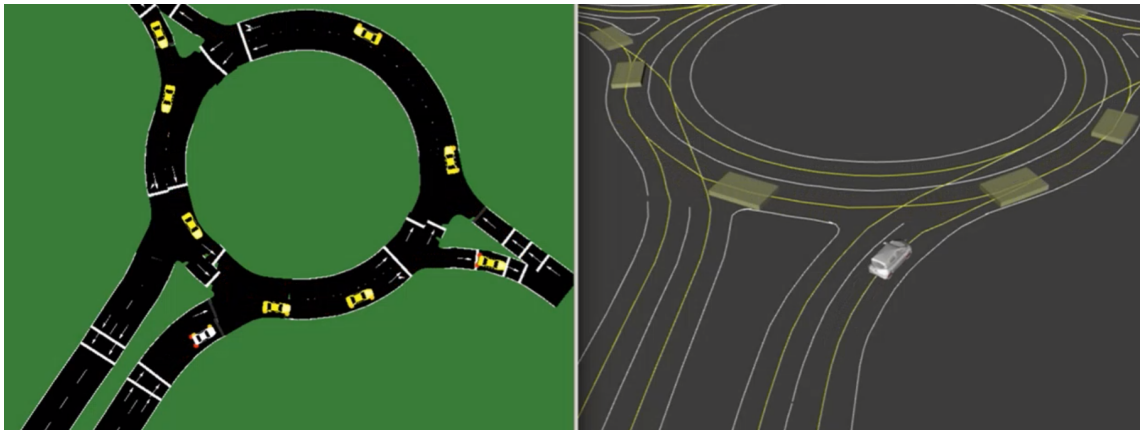


Figure 3.12: The coupled simulator. The AD vehicle is in gray, while all the MD vehicles are in yellow. Note that the roundabout in SUMO has been designed using the HD map representation of the roundabout shown in Figure 2.1 .

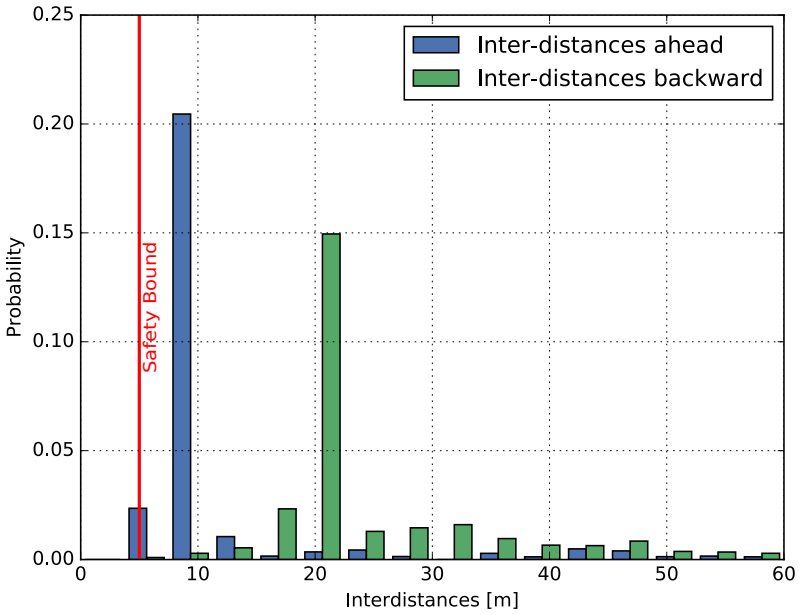


Figure 3.13: Inter-distance distributions w.r.t. the vehicle ahead (blue) and behind (green) during an insertion maneuver. The red line represents the 5 m safety gap.

Table 3.1: The average insertion time, the percentage of average waiting time relative to the nominal case and the average number of insertions as function of vehicles flow for the single-lane roundabout case.

Flow size	50	75	100	125
Crossing Time (s)	5.60 (+1.3)	7.32 (+1.7)	10.05 (+2.39)	15.26 (+3.63)
Number of Insertions	24	21	18	16

Moreover, it is also interesting to quantify the insertion rate w.r.t. the vehicles flow. Table 3.1 illustrates the average insertion rate and waiting time as a function of the number of vehicles in the flow. The insertion time is computed considering that the total length of the decision and transition zones is 33.4 m and the nominal speed in this zone is 30 km/h, which gives us a nominal waiting time of 4.2 seconds.

As one can see, the number of insertions decreases w.r.t. the number of vehicles in the flow. Conversely, the waiting time increases.

3.4 Two-lane Roundabout Crossing

3.4.1 Problem Statement

Let us see in this section how to exploit the virtual vehicles to model the behavior of other vehicles to cope with nudging behaviors. Again, the roundabout shown in Figure 2.1 has been considered. Remember that according to the navigation strategy, an AD vehicle crosses a roundabout only by following the outermost lane (and therefore cannot change lanes). One of the most difficult issues in this scenario is the handling of vehicles that perform lane change maneuvers from the inner ring of the roundabout to outer one which is the one that the AD vehicle uses in this method. In practice, it is very challenging to predict a lane change maneuver, especially when vehicles attempt to make a lane change with nudging [55]. Three different strategies to handle the navigation of multiple vehicles inside a two-lane roundabout have been proposed in the following.

To do that, the concept of virtual vehicles is leveraged again as it has been done before to predict the intention of other vehicles. The main idea is to generate an extra virtual instance of a given vehicle on the innermost lane of the roundabout to occupy the outermost lane according to occupancy methods.

The first method consists in occupying systematically both lanes of the roundabout ring if at least one lane is occupied. This means that, if a vehicle is occupying the innermost lane, the outermost one will result occupied too. Figure 3.14a illustrates this method. The second method consists in occupying the lanes only when there is a significant physical occupancy of a vehicle. This implies that during a lane change maneuver there is always one occupied lane at most, *i.e.,* the outer lane becomes occupied only when the first half of the vehicle has already crossed the bound between the two lanes of the roundabout (Figure 3.14b). The third method occupies the outer lane only when the intention of a vehicle to change lane is detected. This approach is shown in Figure 3.14c. In this case, we assume to have a system that is able to predict when a driver decides to change lane, *e.g.*, by detecting blinkers or lateral distance from the lane center.

In summary, the three aforementioned methods [67] generate an extra virtual instance according to the following statements:

1. Occupying systematically both lanes of the roundabout ring if at least one

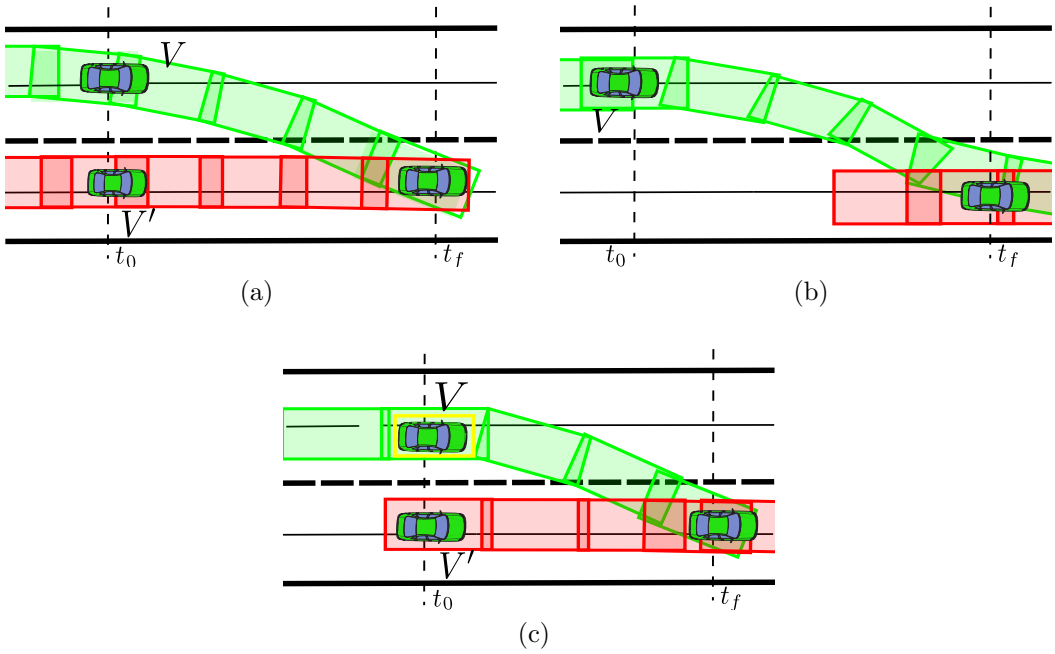


Figure 3.14: Illustration of the three strategies to handle lane change maneuvers in a two-lane roundabout. The vehicle trajectory is green and the corresponding lane occupation is red. Note that for methods 3.14c the lane occupation becomes red when the intention to change lane is detected (yellow square).

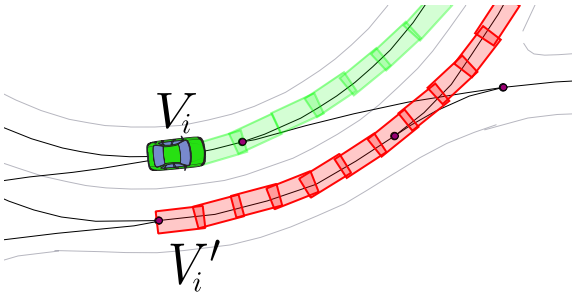


Figure 3.15: Illustration of the strategy to handle the lane change maneuvers in a two-lane roundabout. The vehicle trajectory is green and the corresponding lane occupation is red.

lane is occupied.

2. Occupying the lanes only when there is a significant physical occupancy of a vehicle.

3. Occupying the outer lane only when the intention of a vehicle to change lane is detected.

3.4.2 Simulation Results

For every strategy listed hereafter, several simulations on SUMO with a randomly generated vehicle flow on the full roundabout (two lanes) have been carried out. To properly model a lane change maneuver inside SUMO, a nonzero duration for lane changes has been chosen and the other hypotheses as done in [48] will be still valid. Furthermore, a kinematic model to model the AD vehicle dynamic and to simulate

Table 3.2: Probability of a safety bound violation (ahead and behind), crossing times and effective waiting times for the three methods.

Method	behind	Ahead	Cross time	Wait time
Two lanes occupancy	0%	0%	23.3s	14.46s
Only one lane occupancy	30%	29%	8.6s	4.7s
Intention detection	0%	4%	16.4s	7.79s

the controller part is used. Concerning other vehicles, motion models proposed by SUMO will be exploited combined with the method proposed in section 2.4.2, tuning the parameters of the simulator in a way that the interactive traffic behaves as much realistic as possible to react to AD vehicle behavior and provide more realistic results. In order to quantify the performance of our approach, Figure 3.16 and 3.17 show respectively the inter-distances w.r.t. the vehicles ahead and behind for the three strategies. The first method always ensures safety. For the other two, it is not the case. In fact, the third method violates the safety constraints in both forward and backward cases. This is due to the late detection of lane changes. As a consequence, it can happen that the ego vehicle performs an insertion maneuver when another vehicle has already decided to change lane. In this situation, two scenarios can happen: the ego vehicle forces the other car to change its intentions or the other car completes the lane change despite the presence of the ego vehicle. In the first case, this driving behavior is called nudging. In the second case, the other vehicle cut off the road of the ego vehicle, resulting in a hazardous maneuver or on a collision. The third case behaves as a compromise between the two.

To analyze this aspect, Table 3.2 shows the collision probabilities, average insertion times and average waiting times for the three strategies. Logically, if one wants to have safety ensured, the average waiting time increases. This is due to the fact that, with the aforementioned methods for a double lane roundabout, the available space slots where the AD vehicle can insert are reduced because, in two cases at least, a vehicle matched in the innermost lane produces also an occupancy on the outermost one. As a consequence, the capacity on the roundabout ring is more reduced w.r.t. a single lane one. The first method (Figure 3.14a) tends to be too overly-conservative. Conversely, the second method (Figure 3.14b) is much more aggressive because of the lack of lane change prediction. However, this method has a large unsafe set of configurations. Finally, the third method (Figure 3.14c) behaves as a compromise between the two.

These three approaches have been compared in terms of both safety and system availability and the results are reported in section 3.4.2. However, in order to use proficiently such method, a precise lane change intention detector is required. In this work, we limit our approach to the methods based on statement 1 (Figure 3.15) because such lane change intention predictor is not available in our system architecture and our roundabout test bed is composed of only a one-lane roundabout.

3.4.3 Real Experiments

To test the whole system architecture in a more realistic scenarion, the whole pipeline has been installed on an autonomous Renault Zoé using the ROS middleware. More details about the system architecture of our experimental cars are provided in [99].

We have tested the roundabout insertion first in a hybrid environment (*i.e.*, with simulated vehicles moving on the test track) then with real road agents detected with a LiDAR-based perception system. In the experimental autonomous car, the

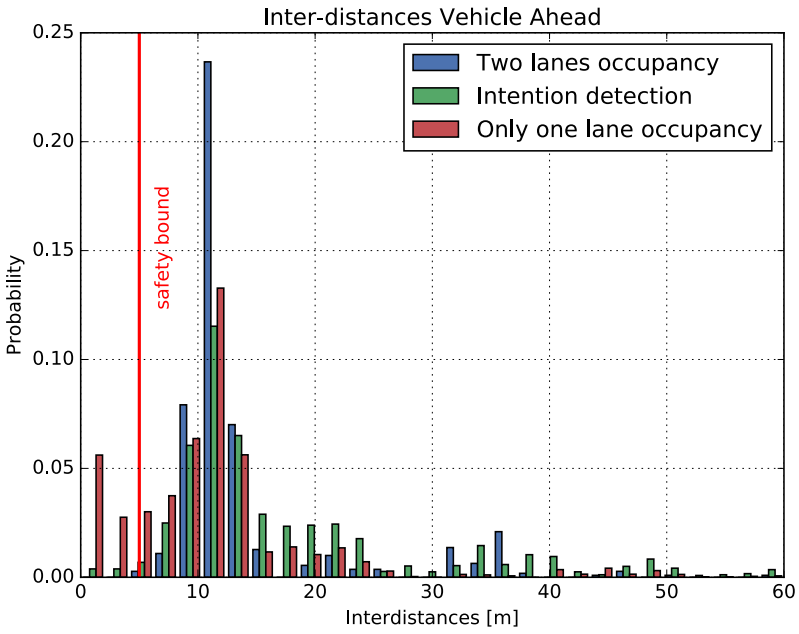


Figure 3.16: Inter-distances distributions w.r.t. the vehicle ahead for the three strategies.

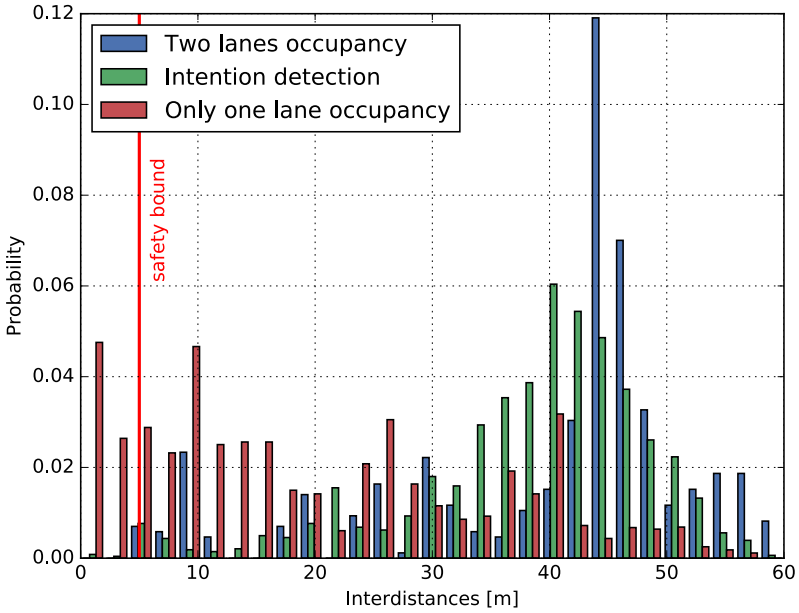


Figure 3.17: Inter-distances distributions w.r.t. the vehicle behind for the three strategies. One can see that the intention detection method behaves as a compromise between the other two.

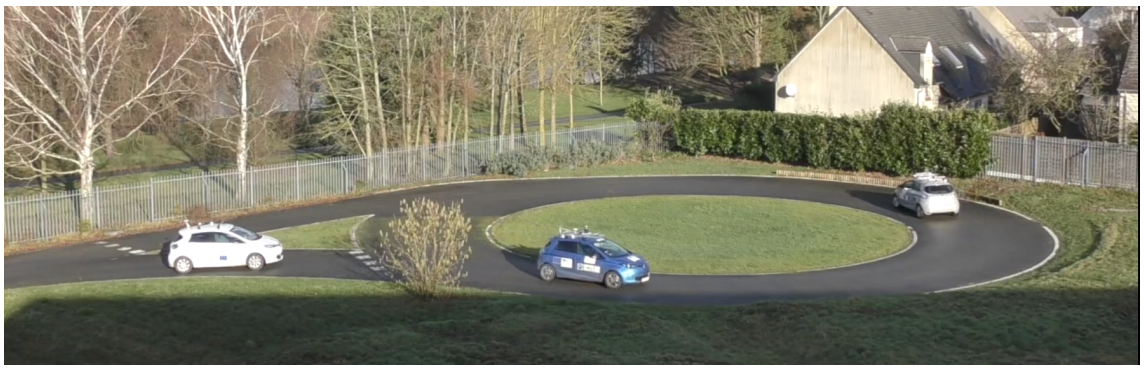


Figure 3.18: The experimental circuit "Seville" and the experimental Renault Zoés used during the real outdoor tests. In our configuration, the white car (fully autonomous) is the AD vehicle, while the others are the road users (manually driven).

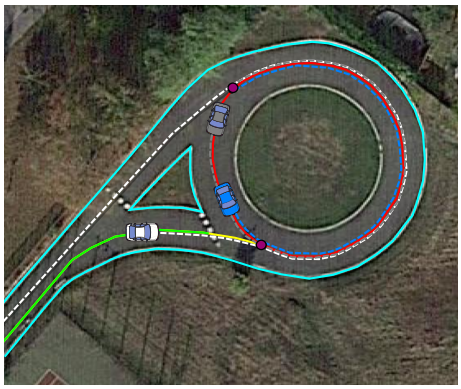


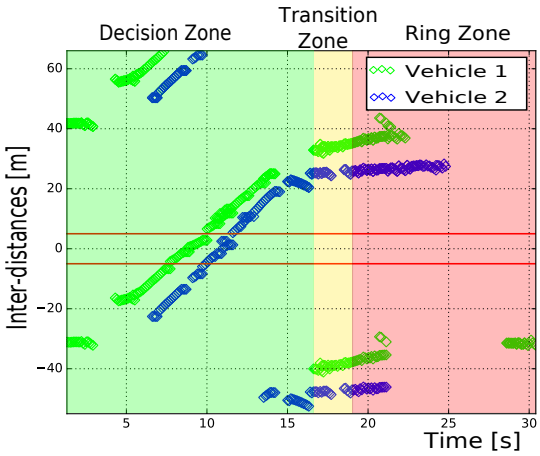
Figure 3.19: The experimental circuit "Seville" with the corresponding HD map and the experiment configuration depicted in Figure 3.18. The AD vehicle is blue and the MD vehicles are green. Notice that the situation case described in section 3.3.3 may occur.

system controls the throttle, the brake pedal and the steering wheel. In our case, the longitudinal motion of the vehicle (*i.e.*, acceleration and brake) is computed according to the traffic situation. In particular, based on the algorithm explained in section 3.3.6, the vehicle can either perform an insertion maneuver into the roundabout or decrease its speed to let other cars go ahead, eventually effectuating a stop at the give-way road sign. Regarding lateral motion, a simple lane keeping is performed and for the detection part, we use a state-of-the-art LiDAR object detection algorithm [13] able to provide information about the detected objects in the form $V_i = [\underline{s}_i, \overline{s}_i, v_i, N_i]$. Note that N_i and the curvilinear conversion have been computed according to previously explained methods. To better illustrate the experimental scenario, Figure 3.18 illustrates this situation.

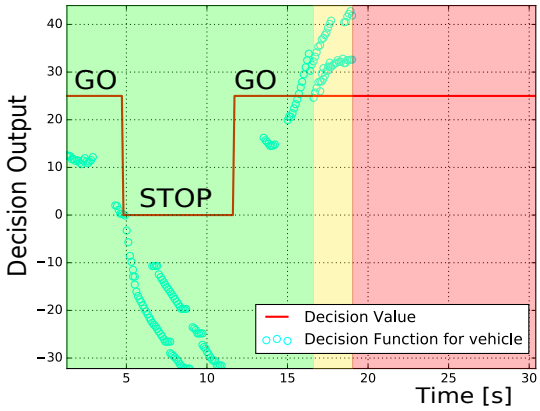
Let us consider a scenario where two cars drive close to each other inside a roundabout doing infinite loops and the AD vehicle has to enter the roundabout (Figure 3.19).

Figure 3.20a illustrates the (virtual) inter-distances of the AD vehicle w.r.t. the other road agents (backward and ahead) accordingly to the three zones. As one can see, the sign of such distance depends on the relative positioning between the MD vehicles and the AD vehicle.

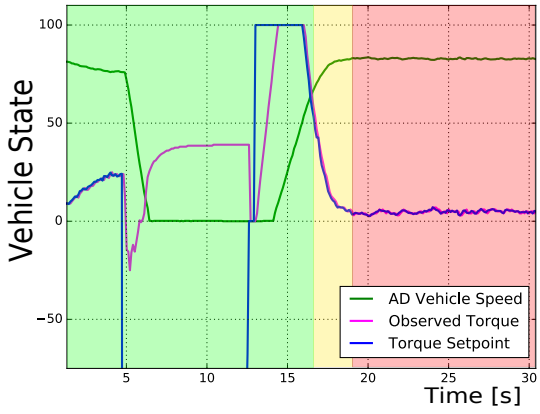
Notice that due to the concept of vehicle instances a vehicle inside the roundabout can be both ahead and behind the AD vehicle, which is coherent with the circular shape of the roundabout. Furthermore, the colors of the background denote the



(a)



(b)



(c)

Figure 3.20: Experimental results. Figure 3.20a illustrates the (virtual) inter-distances of the two cars w.r.t the AD vehicle as a function of time. Notice that the same vehicle may appear both behind and in front of the AD vehicle because of the vehicle instance concept. Figure 3.20b depicts the value of the decision function (red) and the result of Equation 3.23 (blue dots) for every vehicle (virtually) behind the AD vehicle (points < 0 in Figure 3.20a). Figure 3.20c shows the torque setpoint and the corresponding vehicle torque and speed (amplified by a scale factor) as a function of the decision taken in Figure 3.20b.

zone of the roundabout where the AD vehicle is. As one can see, during the crossing of the transition and ring zones, the safety gap is always kept.

If we observe carefully the points in the decision zone, one can see that the other road agents virtually overtake the AD vehicle. In fact, Figure 3.20b shows that, due to the negative values of the constraint of Equation (3.22), the decision changes from go to stop. As a consequence, the AD vehicle lets the other vehicle go ahead and enters into the roundabout behind them.

To better understand the situation at decision and control level, Figure 3.20c shows the system behavior during the whole maneuver. In particular, one can see that the set-point torque changes following the decision output in Figure 3.20b. Consequently, the vehicle speed and the applied torque to the engine change accordingly.

Finally, when the red function decreases to zero in Figure 3.20b, the controller performs a safe stop maneuver. Conversely, once the decision-making part decides to let the vehicle enter in the roundabout, the controller accelerates accordingly. Note that when the vehicle is completely stopped (*i.e.*, $v = 0$) it is still in the decision zone. This means that no safety violation occurred during the maneuver.

3.5 Conclusion

In this chapter, we have studied an adaptation of the virtual platooning concept to the roundabout crossing problem. This idea has the advantage to be easy to implement in an embedded system that exploits a map-based approach. This work has also shown the importance of exploiting a map to model a roundabout since all the calculations are done in a curvilinear framework. As proven, the algorithm has no deadlock. It has also been proposed a strategy to handle situations where both autonomous and manually driven vehicles are involved. Thanks to a safety diagram the algorithm performance has been evaluated in terms of safety.

Then, an extension of this strategy to real-word traffic has been proposed to handle the uncertainty of localization and the issues linked to the detection and tracking of uncertain objects. Furthermore, uncertainties in regular vehicles intentions, which are hard to predict and understand too, have also been included in the proposed approach. Finally, traffic regulations, such as the right of way, have been added to the framework to make this approach effective in a real-life scenario. It has been shown how the virtual instances of vehicles can be used to handle not only the MD vehicles unknown intentions, but also the particular shape of roundabouts. Then, an approach with occupancy intervals to compute the best gap to fit during a roundabout insertion maneuver has been proposed too. The choice of representing objects occupancy with intervals has been made to include the size of MD vehicles and a possible uncertainty about their estimated occupancy. This approach has been tested under a simulated traffic flow generated from real data. The degree of interaction of the generated flow has been used to re-create a scenario close to real world driving. It has been demonstrated that the proposed insertion maneuver ensures safety. Moreover, some performance indexes to evaluate its efficiency in terms of traffic fluidity have been proposed.

To handle the problem of a double-lane roundabout, a lane change intention detector is required to obtain safe and not overly cautious performance. If this technology is unavailable, the authors suggest using instead a worst-case occupation method that always provides a safe insertion. Moreover, if a lane change detector is able to detect also nudging, one can discriminate between a real intention of a driver

to perform a lane change and a false alarm. This approach should add efficiency in the insertion maneuver, decreasing waiting times without compromising safety.

Finally, the proposed method has been tested on an experimental test circuit with real road users and a real AD vehicle in order to evaluate the performance of the proposed algorithm in a real-world scenario with a perception system that provides information about the surrounding road agents.

4 LiDAR-based Road Users Detection with Map Filtering and Uncertain Localization

4.1 Introduction

The aim of this chapter is to study a LiDAR-based perception system to allow an AD vehicle to perform an insertion maneuver in a roundabout with a MD vehicles flow inside it. The main sensor that has been used to achieve on-board perception is a 360° LiDAR sensor that revolves around a fixed axis and provides scans about the AD vehicle surrounding environment. The choice of implementing such a task without exploiting any vehicle to vehicle (V2V) or vehicle to infrastructure (V2I) communication is justified by the fact that V2V communication between road users is not always available. In fact, even if there is a wide set of works in the literature that focus on V2V communication and cooperative methods, nowadays there are a few numbers of MD vehicles that have such technology truly included in their on-board systems. On the other hand, V2I offers an appealing solution to improve the knowledge about the surrounding driving scenario exploiting intelligent infrastructure and Road Side Units (RSU) to broadcast information to the AD vehicles. However, such intelligent infrastructures are seldom available in public roads because of their costs to be installed and maintained.

First, a focus on the information needed to properly feed the decision algorithm explained in the previous chapter has been done. Specifically, the interest is in providing a compact, accurate and consistent information about detected road users. This is crucial to implement a safety-critical driving maneuver as for roundabout navigation. To prevent accidents and vehicle collisions, objects occupancy estimation has to be precise as much as possible. To achieve such a task, one needs to handle the uncertainty of the localization of the AD vehicle. This is necessary to obtain a consistent and robust representation of the driving environment.

Then, it has been also proposed to include the information provided by a HD map directly into the perception process. Indeed, performing object detection by the fusion of LiDAR data with HD maps information can help in both improving the detection performance and allowing faster computations for a real-time use-case. To do so, the HD map information is exploited to detect only road users (i.e., the carriageway) disregarding every other object that is not on the drivable road surface. Regarding pedestrians, their presence has been considered only when they enter in the drivable road surface. The prediction of pedestrians intentions (e.g. predict if a certain pedestrian is about to cross a pedestrian crossing) is out of the scope of this dissertation.

The result of such steps have to be converted into curvilinear occupancy to be given as input to the decision module. In order to be consistent with the formalism presented in the previous chapter, HD maps are again exploited to provide an occupancy interval relative to the HD map. This approach allows to consider objects

occupancy as mono-dimensional occupancy intervals rather than to have information regarding the space occupied by a detected object in terms of global volume and area. This approach has several advantages. First, it allows to have a compact representation about the along-track occupancy of the MD vehicles. Secondly, this approach allows not to compute explicitly the heading angle of perceived objects (that is known to be a non-trivial task for LiDAR-based perception) as the along-track occupancy is projected along HD map polylines regardless of how the vehicles move. This work has been done in collaboration with Edoardo Bernardi who was supervised during his internship by Prof. Philippe Xu and I. Some results about this work have been published in [12].

This chapter presents a perception pipeline that takes as input a 3D LiDAR point cloud, a localization system and a HD map and returns the lane-level curvilinear occupancy of the road users. It is organized as follows. In section 4.2, overview of LiDAR-based perception methods is provided, differentiating machine learning based approaches as opposed to geometry-based ones. Then, the 3D point cloud segmentation method proposed by Zermas *et al.* [101] is detailed in section 4.3. This algorithm has been used to process the raw LiDAR data and serves as input to the rest of the pipeline. In section 4.4, a novel approach to compute the 2D spatial occupancy of detected obstacles is introduced, taking into account uncertainty of the vehicle localization. Then, this occupancy is used to filter out obstacles that are not lying on the road. Finally, the lane-level curvilinear occupancy is computed from the remaining detected road objects. In section 4.5, the concept of integrity in this case study has been defined both from a Cartesian 2D occupancy and lane-level curvilinear one. Finally, experimental results are reported in section 4.6 to evaluate the approach.

4.2 Lidar-Based Perception State-of-the-Art

The field of LiDAR based perception for self-driving vehicles has been widely studied in the recent years and different techniques and methods have been developed. The aim of this section is to explore the existing literature about LiDAR based perception for road users detection. This review will be useful to understand the advantages and drawbacks for each proposed method, allowing us to choose the method that better fits our specifications for our use-cases.

In this section, it has been decided to group the existing literature in two different groups. The first one corresponds to methods that exploit machine learning based techniques, as for example the use of artificial neural networks to treat LiDAR raw data and obtain a representation of the perceived environment. On the other hand, the second group does not rely on machine learning but exploits geometric information provided by different sources (i.e. LiDAR data, road shape, etc.) to build a representation of the obstacles around the AD vehicle.

4.2.1 Machine Learning-based Techniques

One of the mostly recent approaches to obtain a representation of the road obstacles from LiDAR data is the use of machine learning based algorithms in combination with artificial neural networks. The application of such techniques in AD vehicles perception has started to largely spread in the last years accordingly to the development of more and more powerful hardware architecture for AD vehicles on-board

embedded computers. The main objective of these techniques is to utilize a pre-trained network to extract relevant information about the traffic participants. Such information can be vehicles classification and their occupancy. For the last one, such methods are able both to estimate it from acquired data and to infer them from their pre-trained neural networks. In the following part of the work, we are going to analyze the main state-of-the-art algorithms of this category.

As said before, neural networks based perception can be used for several different tasks. For example, they can be used to implement a semantic segmentation on the acquired LiDAR point cloud data. This step consists in assigning to each LiDAR point a class (e.g. ground or non-ground) as performed for images pixels. As it has been shown by Mei *et al.* [70] this operation can be performed by considering the LiDAR point cloud as a range image. Such an image is obtained by projecting the 3D LiDAR points onto a 2D image based on the distance from the sensor. Once this step is done, the image is processed with classical image-based computer vision methods. Alternatively, this problem can be addressed at a point-wise classification problem directly on the LiDAR 3D point cloud [96]. Moreover, some other semantic segmentation methods for objects detection based on deep learning methods can be found in [35, 71] and [39].

Some popular neural networks that can be found in the literature to perform vehicles detection and bounding boxes estimation are VoxelNet or Fully Convolutional Neural Networks [56, 104]. Once vehicle detection and bounding boxes estimation have been achieved, machine learning-based approaches can also be employed to perform the comprehension of dynamics of moving objects, as for example the inference of MD vehicles intentions [91] and their tracking and speeds estimation [34, 83].

Machine learning-based techniques are well suited for object classification purposes. In particular, considering the acquired LiDAR point clouds, it is possible to identify different categories of road agents as cars, trucks, bicycles, bikes, pedestrians or not moving objects such as trees or panels. This has several advantages. Firstly, this process can be helpful during the tracking process, because if we have the information about the class of every object, it is possible to choose a correct evolution model that better fits the object dynamics, providing an overall improvement of the tracking performance. Then, the class information about perceived object can also be helpful in rejecting objects that are not relevant for a specific use-case, for example buildings, trees or static object in the case of AD vehicles navigation.

However, some drawbacks can be identified in these methods such as the necessity of a labeled dataset to perform training of the neural network. In this case, the training dataset should contain a heterogeneous variety of driving scenarios and behaviors, in order to provide the network as much information as possible regarding possible driving behaviors. As a consequence, one key issue of machine learning based techniques is that obstacles that are not represented in the training dataset will not be detected. For this reason, an obvious but uncommon obstacle, *e.g.*, a cow, is likely not to be detected (or detected wrongly). This may lead to critical safety issues that will not appear if we use geometric methods.

In this work, a geometric method to perform road obstacles detection has been implemented rather than a machine-learning based one because, even if these methods offer a more sophisticated solution and their performance is increasing more and more over the years, we obtained satisfactory performance for our goals from the use of geometry-based ones. Nonetheless, in the particular context of this work and the research project related to it, we disposed of an on-board architecture with a limited computational power on the test vehicles. Such a constraint forced us to

choose more optimized and viable methods than machine-learning based ones for real tests implementations. Furthermore, even if there exist in the literature some machine-learning based methods that can perform in real-time with a reasonable consumption of our computational power, we believe that the method adopted in this work provides satisfactory performance for our use-case.

4.2.2 Geometry-based Techniques

Another approach that is commonly used to achieve LiDAR based perception consists in treating perceived environment from a geometrical point of view. In particular, this set of techniques relies in the application of geometric principles to treat point cloud data. In such cases, it is widely common to use a geometric concept to discriminate LiDAR points in a point cloud that satisfies a certain criterion or property or to extract some features according to a certain geometry tool. In general, such techniques are based on a classic set of substeps. Each substep has the goal of implementing a specific treatment on the output point cloud of the previous one. A general straightforward example of such detection pipeline is composed of the following steps.

- **Ground not-ground segmentation**: this step consists in separating the acquired LiDAR point cloud in two different categories: the points that belong to the ground surface, and the ones in which the points do not. The aim of such a step is to discard points that are classified as ground points in order to reduce the size of LiDAR points to treat in the following steps. This is useful for road obstacles detection. However, on the other hand, the dual reasoning can be performed on ground points if one is interested in extracting the drivable surface rather than road obstacles.

- **Clustering**: LiDAR points are processed to group them according to some similarity criteria, as for example for sets of points that are most likely to be part of the same object.

- **Bounding boxes**: clusters are represented by a bounding volume that contains all its points. Such volumes can be boxes or polygons representing the object occupancy.

4.2.2.1 Ground Not-Ground Segmentation

In the literature, there exists a wide set of algorithms that can perform a ground/not-ground segmentation of LiDAR point clouds. Some methods are based on grid approaches. Such methods split the analyzed surface into small cells with fixed dimensions. The overall result is that the whole space is divided by a rectangular grid of cells. Each of the cells is then processed individually by identifying all the LiDAR points overhanging it. Then, every grid cell is classified either to belong to ground or to not-ground. However, the size of the cells has to be carefully chosen in order to obtain a good result. To perform this step Chen *et al.* [21] compute the height variance of points belonging to the same cell, if this value is above a certain threshold, the cell is classified as not-ground. Another technique by Korchev *et al.* [50] consists in computing two different maps: one that contains the maximum height of a cell point and the other one the minimum one. The aim of this is to compute the difference between the two maps and a thresholding technique is deployed to assign the correct class to each one of the grid elements. The same

grid-based reasoning can also be performed in polar coordinates rather than 2D Cartesian space. Here the goal is to exploit circular geometry and polar coordinates to build a radial grid, in which each radial cell is obtained as a portion of a circular sector. The previous methodologies are able only to perform a binary classification of each cell. For this reason, Asvaldi *et al.* [3] utilize a 2.5D grid instead of a normal occupancy one. The main advantage of this is that this approach allows to insert in each cell the average height of all LiDAR points and reject the points with this value below a threshold.

Another commonly used technique for ground/non-ground segmentation of LiDAR points is a beam based technique that consists in splitting the circular area scanned by the LiDAR into small circular sectors and then analyze each sector to extract points not belonging to the ground surface.

Following this approach, Himmelsbach *et al.* [63] propose a method in which each circular sector is split into several bins, based on the radial distance from the LiDAR sensor. Acquired LiDAR points are assigned to a specific bin according to their position and distance from the LiDAR sensor. For every bin, the point having the minimum z-coordinate is chosen as a prototype point. This point has the role of providing a compact representation of all the points contained into the bin. This point and all the others of the same sector are then used to estimate the best line that fits all the prototypes of that space portion. The same procedure is repeated for all the sector and the set of obtained lines are used to represent the ground plan. Such lines are then exploited to classify all the acquired LiDAR points into ground or not-ground based on a likelihood criterion.

A similar methodology is presented by Chu *et al.* [22]. This approach, however, consists in batch processing all the LiDAR points generated by the sensor at each azimuth angle. These points are then classified as ground or not-ground considering different factors such as the slope of the line conjoining two consecutive points, the height difference between two conjoined points or the radial distance from the sensor.

Another method consists in iteratively computing and fitting with the LiDAR points a parametric representation of a plan. Such plan is estimated from a geometric representation and refined at every time step accordingly to the new segmentation results. This approach has been proposed by Zermas *et al.* [101]. In this work, the area scanned by the LiDAR sensor is divided in one or more sub-plans along the vehicle driving direction in order to include multiple plan fitting for the eventual presence of slope changes.

After that, a threshold on all the LiDAR points is performed according to their distance to the plan and the points are finally classified into ground or not ground.

4.2.2.2 Clustering

The goal of clustering methods is to group LiDAR points that belong to the same object in the same cluster. This technique is useful to identify LiDAR points that constitute physical objects inside the detected LiDAR point cloud. The main idea is to assign a class label to each LiDAR point. Consequently, LiDAR points that belong to the same cluster will be identified with the same label.

In the literature, there exists a wide range of algorithms that perform clustering. In particular, if a 2D or 2.5D grid has been used to implement a grid based segmentation method, it is possible to apply an algorithm that aims to find connected components in a binary image to perform the clustering step. In such a case, the binary image is given by the grid computed before, where the values 1 and 0 represent

the two classes ground or not ground. One algorithm that exploits such technique is the run-based two scans [53] algorithm. In this algorithm, a grid is traversed twice to find the connected components. The first pass is made to get the linked components and to assign them a class label, while the second pass is used to merge linked components that are close enough to represent a single object.

By using the same data structure, it is also possible to implement a method for moving objects detection by comparing the estimated grid with a map populated with the last obtained observations [3].

On the other hand, if one applies a segmentation approach to LiDAR points, a point cloud is returned instead of a grid. From such a representation, it is possible to extract a range image and to use image-based techniques to perform clustering. The 3D LiDAR point cloud is first projected onto a 2D cylindric range image and each not-ground point is treated as an image pixel. In the approach proposed in [101], Zermas *et al.* treat this image identifying connected components by an adapted two runs labeling algorithm. On the other hand, Bogoslavskyi *et al.* [42] use a technique that involves the angle between the line formed conjoining two adjacent pixels and the line conjoining the sensor positioning to implement a criterion for defining if two adjacent points belong to the same cluster or not.

Finally, another approach consists in using a radial bounded nearest neighbor technique [47]. Such an algorithm assigns the same label to LiDAR points that are within a certain radial neighborhood of a given LiDAR point.

4.2.2.3 Bounding Boxes

This step has the goal of creating bounding boxes or bounding polygons that encapsulate the clustered points obtained at the previous step. The main objective of building a bounding polyhedron is to identify at a geometric level the volume occupied by an obstacle or by a MD vehicle. Furthermore, this compact representation of a clustered object allows to forget about single LiDAR points and to treat the more compact bounding polygon directly instead.

A first simple approach to create 3D bounding boxes for clustered objects is the use of the Principal Component Analysis (PCA) algorithm. This technique computes on the three x, y and z-axes the principal components of a given cluster of points. These directions are then used to compute the sides of the 3D bounding box by finding the maximum and minimum values of the clustered LiDAR points w.r.t. the directions of the three principal components. However, in this approach there are many drawbacks. The main drawback is that the estimated orientation of the object is computed according to the principal components orientation. In other words, such orientation is heavily influenced by the shape of the cluster. Some typical LiDAR cluster shape that can lead to poor performances are L-shapes, obtained by a three-quarter view of a car-like object.

In order to cope with such problem, several techniques have been developed in the literature. The first technique consists is the L-shape fitting method. Such method has been detailed in Zhang *et al.* [97]. In this work, the authors propose three different criteria to extract the extreme points used to compute bounding box vertices. In order to find the best configuration for the L-shape fitting, an area minimization optimization problem is solved according to the different optimization criteria. In other words, a point-to-edges squared error minimization process is performed. This results in a bounding box estimation with small orientation error. However, it requires a longer computational time.

Another example that exploits the localization of L-shape vertices and its consequent shape fitting via optimization problems are also considered by Qu *et al.* [79].

One other approach consists in estimating the object occupancy on both the x and y-axes by computing the minimum area rectangle that contains the given set of LiDAR points. One of the mostly used techniques to achieve such a task is the computation of the minimum convex hull that surrounds all the points of a specific cluster. Once this step is achieved, for each side of the obtained polygon a rectangle having a side parallel to the polygon side is computed. Finally, once this computation has been performed for every side of the polygon, the rectangle having the minimum area is identified. Börcs *et al.* proposed an $O(n^2)$ approach that exploits such technique [16], where n is the convex hull number of vertices. Another approach shown by Eberly implements an $O(n)$ bounding box method based on rotating calipers [17].

4.3 3D Point Cloud Segmentation for Object Detection

4.3.1 Ground Not-ground Segmentation

In this first block of the detection chain,the points of the raw data input LiDAR point cloud are separated in two subsets: the ground points and not-ground points. The aim of this phase is to reduce the number of points to be processed in the next steps. In particular, in this work only road obstacles detection will be considered. For this reason, only the non-ground points are forwarded to the next step.

By making this choice, it has been implicitly considered that the ground and non-ground classification is correct for every point and that vehicles have a sufficiently large height to not include all the points completely that represent a vehicle object inside the ground points. In such a way, there is guarantee that no vehicle objects will be classified as ground points, avoiding the missed detection of the vehicle.

Moreover, in this work it has been also assumed that the road surface is flat, i.e. there are no ramps or any other kind of slopes. To cope with this hypothesis, one needs to point out that the algorithm presented hereafter can be tuned to perform ground plan fitting for non-flat surfaces. However, this aspect is out of the scope of this work and it is not treated in detail.

On the other hand, for the sake of completeness, one needs to mention that there also exists approaches that aim to estimate the free space (i.e. the navigable zone for an AD vehicle) rather than focusing on estimating road obstacles.

As it has been previously shown in the state-of-the-art section 4.2, we investigate the main existing approaches to find the most suited method for our architecture. To the best of our knowledge, it has been found out that the plan fitting algorithm proposed by Zermas *et al.* [101] behaves as a good compromise between precision and computational time.

Moreover, such algorithm satisfies the hypothesis that have been listed before. In fact, the algorithm is based on the assumption that ground points are identifiable because belonging to plans and the points having small height values are more likely to be classified as ground points.

The algorithm is based on a RANdom SAmple Consensus (RANSAC) algorithm for plan fitting, initialized with a set of random points picked up from the raw-data point cloud. Notice that in many cases a single plan is not correctly representative of

the real ground surface and it cannot describe small changes of slope. Consequently the point cloud can be divided into N_{segm} segments along the x-axis, which represent the along direction of AD vehicle motion. By doing that, the plan fitting is performed on each of them and the resulting ground plan is represented by their union. For a plane estimation, the first plan is estimated by using random seed points. A simple linear model shown in Equation 4.1 is used

$$\begin{aligned} ax + by + cz + d = 0 \\ N^T X = -d \end{aligned} \tag{4.1}$$

with $N = [\ a \quad b \quad c\ ]^T$ and $X = [\ x \quad y \quad z\ ]^T$.

Then the estimation of the normal N to the plane is computed by using the covariance matrix C_i computed by a set of seed points P_{G_i} extracted in the previous step.

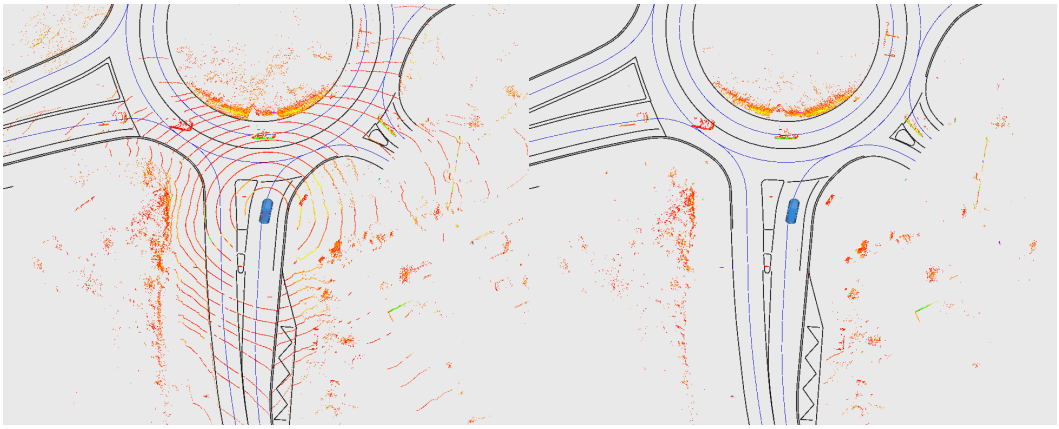
$$C = \sum_{j=1:|P_{G_i}|} (s_j - \hat{s})(s_j - \hat{s})^T \tag{4.2}$$

with $s_j \in P_{G_i}$ and $\hat{s}$ is the mean of all s_j.

Equation 5.5 shows the empiric computation of the covariance matrix. This quantity is used to compute the three singular vectors representing the three main directions of dispersion for the seed points, that will be used to compute N according to Zermas *et al.* [101].

Finally, each point $p_j \in P_i$, where P_i denotes the points of the point cloud, is classified as ground or not-ground by evaluating its distance w.r.t. the estimated ground plan. Once this step is done, the new obtained ground points can be used as new seeds to iterate and refine the plan estimation and points classification for a fixed number of iterations.

Figure 4.1 illustrates a LiDAR raw-data point cloud and the corresponding non-ground point cloud obtained after this step.



(a) (b)

Figure 4.1: The result of the ground non-ground segmentation: Figure 4.1a shows the whole raw data point cloud before the treatments, while Figure 4.1b depicts the point cloud that contains only the non-ground points.

4.3.2 Clustering Algorithm

After the ground removal module, the system needs to extract information about the vehicles that are present in the driving scenario. For this reason, it has been decided

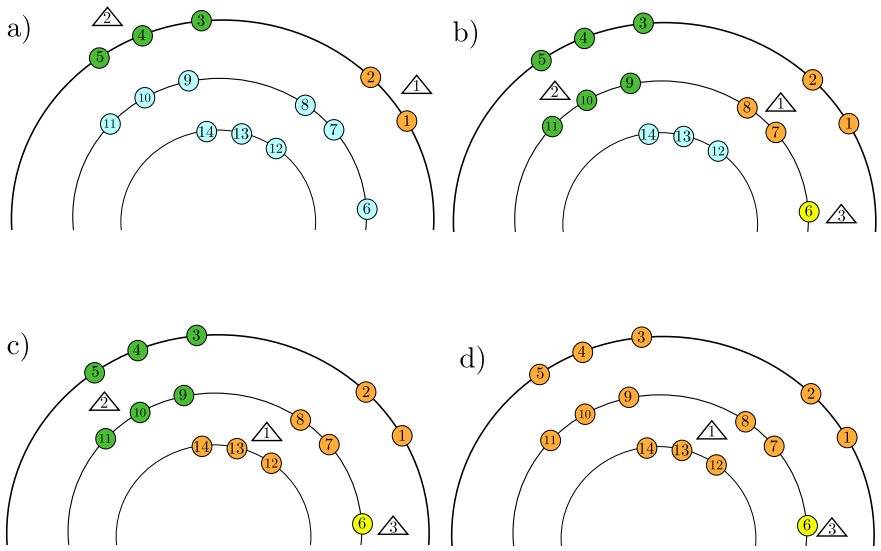


Figure 4.2: Four main steps of the Scan Line Run clustering algorithm. The orange, green and yellow points represent the points classified in the same run, while the blue point represent points that have not been assigned yet. Steps (a), (b) and (c) illustrate the labels propagation and the creation of a new run. Step (d) depicts the merging of different runs. This figure is taken from the paper of Zermas *et al* [101].

to exploit a clustering algorithm to individuate the vehicles in the pointcloud. The goal of the clustering step is to group all the LiDAR points that belong to the same object under the same label. This step is crucial to determine objects occupancy. In fact, once the set of points that belong to the same cluster has been computed, a bounding polygon can be extracted by such set in order to provide a more compact representation of the perceived object.

In order to fulfill the requirements of our system, we investigate again the existing literature to find an algorithm that fits as much as possible the aforementioned criteria about real-time implementation. As it has been shown in Section 4.2.2.2, different approaches have already been proposed for clustering LiDAR points.

Therefore, in this work it has been decided to consider the method proposed by Zermas *et al* [101]. This choice has been made because it is the opinion of the authors that this algorithm ensures both low complexity in clusters computation and fast computational time [101]. However, on the other hand, such an algorithm is sometimes prone to over-segmentation of detected objects. In other words, the algorithm could have a tendency on splitting large detected objects in multiple sub-clusters that belong to the same detection.

The main idea of this approach is that the 3D LiDAR point cloud is considered as a 2D cylindrical image, where the row and column positions of the points in the image are obtained by the different layers organization of the sensor. These points are then labeled by using a two-run connected components algorithm as the one shown by He *et al.* [53]. In particular, in their implementation of the algorithm, the authors exploit the rings structure of the LiDAR sensor to optimize clusters computation.

The main steps of the aforementioned algorithm are depicted in Figure 4.2. In this representation, the blue circles represent the LiDAR points that have not been traversed yet. In particular, sub-step (a) shows that two runs are identified and

labeled with different labels (*orange* and *green*). In sub-step (b), a new label is given to a run consisting of only a point that has no potential candidate label to inherit. On the other hand, the other labels are propagated to new runs in the subsequent scan-line. Sub-step (c) illustrates the case where a run has multiple different neighboring labels to inherit. However, only the smallest one is assigned to the new run. Finally, in sub-step (d), the different labels of neighboring runs are merged.

The output of this block is a set of point clouds $C = \{P_k\}$ with $k = 1 \dots l$. In such sets, each point cloud P_k represents the set of LiDAR points that belong to a given vehicle detection. Each cluster represents an object and a unique label is assigned to every cluster. The set C is given as input to the following step.

4.3.3 Convex Hulls Bounding Polygons

Let us describe how to compute a convex bounding polygon to represent in a compact way a vehicle cluster. In particular, only on 2D bounding boxes computation will be considered in this work. For this reason, and because the main purpose of this system is to extract curvilinear occupancy intervals, the z coordinate of LiDAR points is disregarded.

Given a set of points, as for example a clustered point cloud P_i with at least 3 elements, a convex hull C_i is defined as the smallest convex set that completely contains P_i. To perform this computation, the Monotone chain [103] algorithm has been used, with a resulting complexity of $O(n \log n)$ [24]. The result of the convex hull computation phase is a set of polygons $H = \{C_i\}$ with $i = 1 \dots l$, where l is the total number of clusters.

This step provides only the set H to the following blocks.

4.4 Road obstacles lane-level occupancy

4.4.1 Filtering the Points Thanks to the Map

In order to use the navigation strategy described in chapter 3, one needs to compute the lane-level curvilinear occupancy of the road users with respect to a HD map. The point cloud segmentation method described in the previous section provides a set of convex polygons corresponding to the detected objects expressed in the vehicle frame. A HD map can be used to filter out the obstacles that are not located within the road space. To do so, the AD has to be able to localize itself with respect to the map.

Let ${}^{v}X_p = \begin{bmatrix} {}^{v}x_p, & {}^{v}y_p \end{bmatrix}^T$ be the position of a LiDAR point relative to the AD frame. The LiDAR makes measurements in its own frame but since the geometric transformation from the LiDAR frame to the AD vehicle one is very well known through calibration, it is assumed that the LiDAR points are directly expressed in the vehicle frame. To compute the coordinates ${}^{w}X_p = \begin{bmatrix} {}^{w}x_p, & {}^{w}y_p \end{bmatrix}^T$ of this point in the world frame, one needs to use the pose ${}^{w}X_e = \begin{bmatrix} {}^{w}x_e, & {}^{w}y_e, & {}^{w}\theta_e \end{bmatrix}^T$ of the AD as follows

$$ {}^{w}X_p = f\left({}^{v}X_p, {}^{w}X_e\right) = \begin{bmatrix} {}^{w}x_p \\ {}^{w}y_p \end{bmatrix} = \begin{bmatrix} {}^{w}x_e \\ {}^{w}y_e \end{bmatrix} + \begin{bmatrix} \cos {}^{w}\theta_e & -\sin {}^{w}\theta_e \\ \sin {}^{w}\theta_e & \cos {}^{w}\theta_e \end{bmatrix} \begin{bmatrix} {}^{v}x_p \\ {}^{v}y_p \end{bmatrix} \quad (4.3) $$

Then one can determine whether a point lies on the road surface or outside of it by counting the number of times the laser beam intersects road boundaries as

illustrated in Figure 4.3. Cases (a), (d) and (e) represent situations where the point is outside the region (odd number of intersections), while cases (b) and (c) represent cases where the point is inside (even number of intersections).

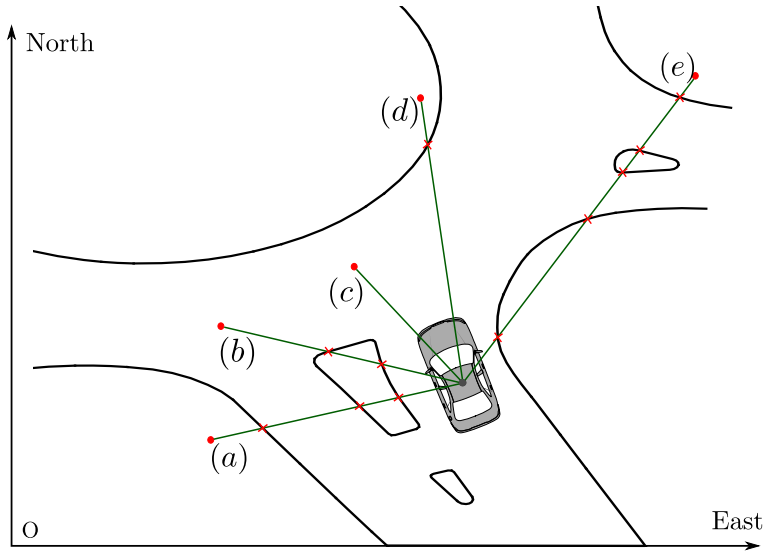


Figure 4.3: Different intersection cases. The drivable zone boundaries are depicted in black and the segments between the AD vehicle sensor and the LiDAR points are in green. Cases (a), (d) and (e) represent situations where the point is outside the region (odd number of intersections), while cases (b) and (c) represent cases where the point is inside (even number of intersections).

The method works well only if the localization of the AD is correct, otherwise erroneous situations may arise as illustrated in Figure 4.4:

1. True Positive: a point inside the region of interest is correctly classified as inside the region of interest.

2. False Positive: a point outside the region of interest is wrongly classified as inside the region of interest.

3. True Negative: a point outside the region of interest is correctly classified as outside the region of interest.

4. False Negative: a point inside the region of interest is wrongly classified as outside the region of interest.

In terms of safety, false negatives (case 4) should be avoided as much as possible as they may lead to hazardous situations. False positives (case 2) are less critical in terms of safety but should also be kept as low as possible in order to increase the availability rate of the navigation system.

In practice, localization is never perfect therefore its uncertainty needs to be carefully taken into account. The main consequence of an uncertain localization is that whether a given point lies on the road space or not may become ambiguous.

In the following, we introduce a method to propagate localization uncertainty onto the occupancy of the detected clusters and show that it is more robustness to non-linear transformations compared to classical linearization approaches. It will be assumed that the localization information of the AD vehicle is provided as a random variable ${}^{w}\hat{X}_e$ following a Gaussian distribution with a covariance matrix P_e.

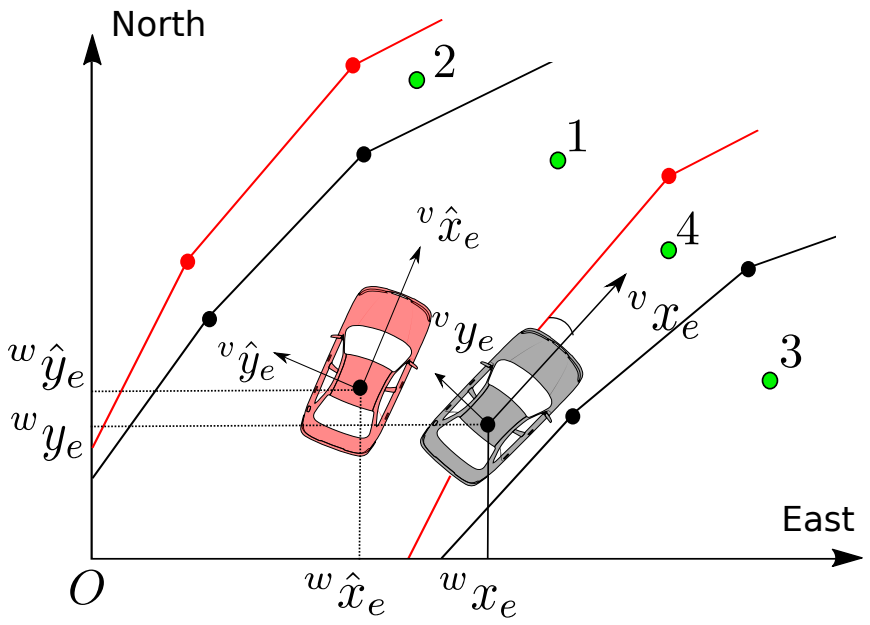


Figure 4.4: Impact of localization errors. The estimate is in red vehicle and the ground truth in black.

We also assume that ${}^{w}\hat{X}_e$ is an unbiased estimator of the unknown true pose ${}^{w}X_e$ of the vehicle, *i.e.*, $\mathbb{E}\left({}^{w}\hat{X}_e\right) = {}^{w}X_e$, or in order words ${}^{w}\hat{X}_e \sim \mathcal{N}\left({}^{w}X_e; P_e\right)$. Such an estimation is typically provided by a Kalman filter.

4.4.2 Linearized Propagation of Point Uncertainty

Let us begin by dealing with the simple case of the propagation of the uncertainty of the pose on a point. Given an uncertain pose ${}^{w}\hat{X}_e$ of the AD vehicle, the position of a LiDAR point in the world frame ${}^{w}\hat{X}_p$ also becomes uncertain as it is related to the AD vehicle pose ${}^{w}\hat{X}_p = f\left({}^{v}X_p, {}^{w}\hat{X}_e\right)$ as expressed in Equation (4.3). Unfortunately, because of the non-linearity of the function f, there is no closed form to describe the probability distribution of ${}^{w}\hat{X}_p$. It is, however, possible to approximate this distribution by a Gaussian one using a first order approximation similarly to an extended Kalman filter.

To do so, the covariance matrix P_p associated to ${}^{w}\hat{X}_p$ will be defined as follows:

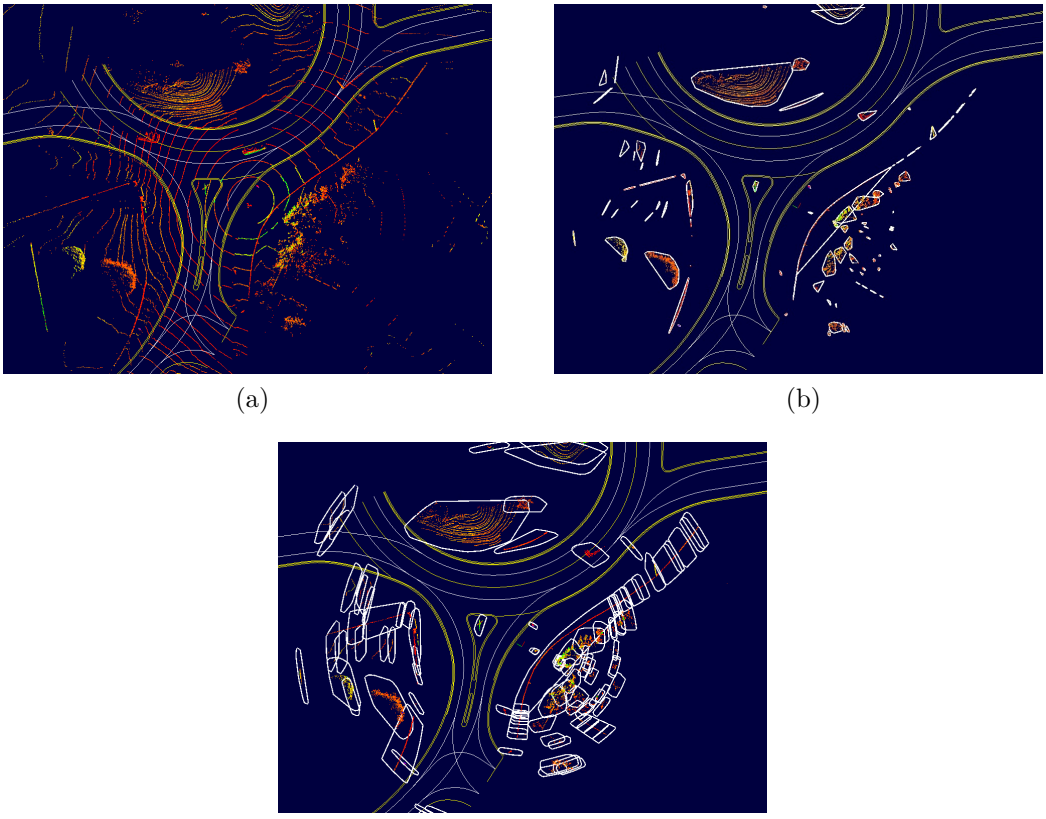
$$P_p = J P_e J^T,$$

where J is the Jacobian matrix of f defined as

$$\frac{\partial f\left({}^{v}X_p, {}^{w}\hat{X}_e\right)}{\partial {}^{w}\hat{X}_e} = \begin{bmatrix} 1 & 0 & -{}^{v}x_p \sin {}^{w}\hat{\theta}_e - {}^{v}y_p \cos {}^{w}\hat{\theta}_e \\ 0 & 1 & {}^{v}x_p \cos {}^{w}\hat{\theta}_e - {}^{v}y_p \sin {}^{w}\hat{\theta}_e \end{bmatrix}. \tag{4.4}$$

The LiDAR point position is now described by a Gaussian probability distribution. The true position ${}^{w}X_p$ of the LiDAR point in the world frame is unknown but, given a risk $\alpha \in [0, 1]$, it is possible to compute a confidence domain $C(\alpha)$ such that

$$Pr\left({}^{w}X_p \in C(\alpha)\right) = 1 - \alpha. \tag{4.5}$$



(a) (b)

(c)

Figure 4.5: LiDAR point cloud processing. Figure 4.5a: Raw-data LiDAR point cloud. Figure 4.5b: Clustering and bounding convex hulls computation on non-ground point cloud obtained at the previous step. Figure 4.5c: Extended convex hulls computed by injecting localization uncertainty .

There exists an infinite number of such confidence domains. In our two-dimensional case with a Gaussian distribution with covariance matrix P_p, this domain is commonly represented in the form of an ellipse with its major and minor axis aligned along the eigen vectors of P_p. This domain is often chosen because it results being the smallest, in terms of area, verifying Equation (4.5). In this specific problem, an ellipse is difficult to be geometrically manipulated. In particular, it is complex to compute the overall occupied space by a perceived object with several points. Consequently, it has been chosen to to use rectangles, which provide a less complex shape to manipulate.

Consider a rectangle having sides aligned with the eigenvectors of P_p and with length l_i computed as:

$$l_i = 2\Phi^{-1}\left(\frac{1+(1-\alpha)^{1/2}}{2}\right)\lambda_i, \tag{4.6}$$

where λ_i is the i-th eigen value of P_p and Φ is the cumulative probability density of the normal distribution $\mathcal{N}(0,1)$. Figure 4.6 illustrates the obtained uncertainty for a given point. It should be noted that the uncertainty is not yet well bounded in this case. Thanks to a set-membership approach, a method to correctly encompass all the uncertainty has been proposed.

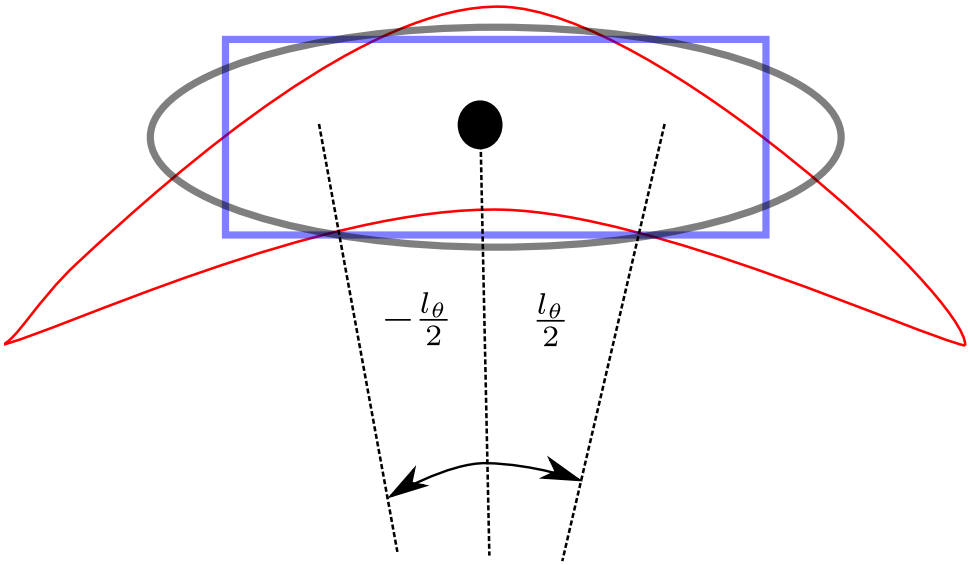


Figure 4.6: Uncertainty ellipse constituting the confidence domain of a point. This polygon has been obtained by applying the linearization method. The true uncertainty distribution is illustrated in red by a banana shape, the black ellipsoid represents the linearized propagation, while the bounding polygon is in blue.

4.4.3 Set-membership Direct Propagation of Point Uncertainty

To overcome the issue seen before, an approach that directly propagates the vehicle pose confidence domain onto the one of the perceived object is proposed. No linearization is necessary.

The covariance matrix P_e of the AD vehicle pose is used to compute the variances in the cross-track (CT) and along-track (AT) directions w.r.t. the estimated heading ${}^w\hat{\theta}_e$ of the vehicle. Let σ^2_{CT}, σ^2_{AT} and σ^2_θ be the resulting variances in these three

dimensions. Similarly to the previous approach, a confidence domain is built over the AD vehicle pose in the form of a "cube" in the CT, AT, and heading three-dimensional directions. The length of each side is given by:

$$l_i = 2\Phi^{-1}\left(\frac{1+(1-\alpha)^{1/3}}{2}\right)\sigma_i, \tag{4.7}$$

where $i \in \{CT, AL, \theta\}$.

To propagate this confidence domain onto a point ${}^{v}X_p = \begin{bmatrix} {}^{v}x_p, & {}^{v}y_p \end{bmatrix}^T$, let us use a set-membership approach. The rectangle confidence domain resulting from the uncertain position is firstly computed:

$$[{}^{v}x_p \pm l_{CT}/2] \times [{}^{v}y_p \pm l_{AL}/2]\,. \tag{4.8}$$

Then, in order to take into account the uncertainty over ${}^{w}\theta_e$, two other rectangles are computed by using a rotation of angles $\pm l_\theta/2$ (in the vehicle frame the heading is always null). In order to represent the space occupied by the rectangle rotation, the circles generated by the farthest vertexes are computed. Then the tangent lines to such circles, passing through the aforementioned vertexes and the corresponding ones in the rotated rectangles are extracted. Finally, the tangent lines intersection points along with the corners of the three rectangles are used to compute a convex hull corresponding to the confidence domain of the point.

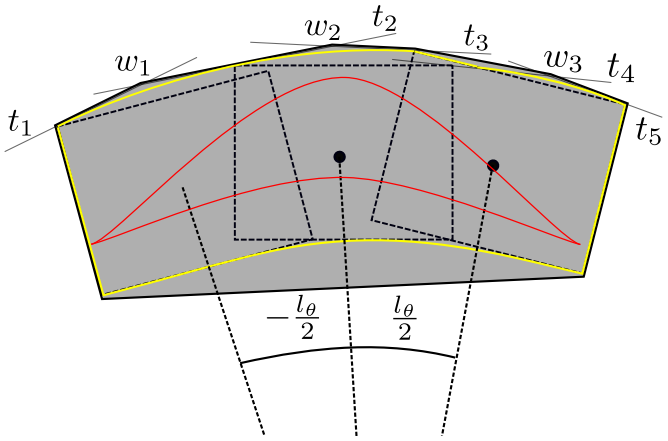


Figure 4.7: Polygon corresponding to the confidence domain of a point (in grey). The space occupied by the rectangle rotation is in yellow. Please note that this convex approximation covers the banana shape illustrating the true uncertainty distribution is red.

This procedure is depicted in Figure 4.7: the rectangles represent the confidence domain resulting from the uncertain position and its rotations. The lines t_1, t_2 and t_3 are obtained as tangent lines to the circle generated by the top left corner rotation, passing through such vertex in the three rectangles. The same reasoning is applied for the computation of t_4 and t_5 by the use of the top-right corner (in a symmetric case also the tangent through the third rectangle vertex has to be computed). Points w_i are computed from the intersections of the previously calculated tangent lines. Finally, the convex hull representing the point confidence domain is computed using rectangle vertexes and tangent lines intersection points.

If one compares this approach to the propagation of uncertainty with the previous one, one can observe that it is more conservative, which corresponds well to our objectives of integrity.

4.4.4 Extended Bounding Polygons of Detected Obstacles

Now knowing how to propagate the uncertainty of pose on a point, we look for a method to apply this concept to the bounding polygons computed in section 4.3.3. Indeed, by considering each perceived object as a polygon, it is possible to propagate the pose uncertainty on every vertex of the convex hull. The overall occupied space is then obtained by computing the bounding polygon of all the rectangles obtained by the uncertainty propagation on the convex hull vertexes. This computation is illustrated in Figure 4.8.

Notice that an alternative to this approach could be to propagate the uncertainty on all the LiDAR points that constitute a cluster and, after that, to compute the convex hull considering all the points that constitute all the polygons obtained from the uncertainty propagation. In this work it has been preferred to reason in terms of polygons rather than a single point to better take into account object shape and to avoid filtering parts of uncertain objects that are outside the road surface.

This technique can be conceptually compared with a Minkowski sum [25] between the general polygon bounding an object and the rectangle approximating the uncertainty ellipsoid of the centroid of such a polygon. The ellipsoid dimension depends on the hull vertex distance from the sensor perceiving it. To observe the overall obtained result until this step, Figure 4.5a illustrates the input raw data, while Figure 4.5b illustrates the output of the clustering step. Finally, Figure 4.5c depicts the extended bounding polygons corresponding to the previously computed clusters.

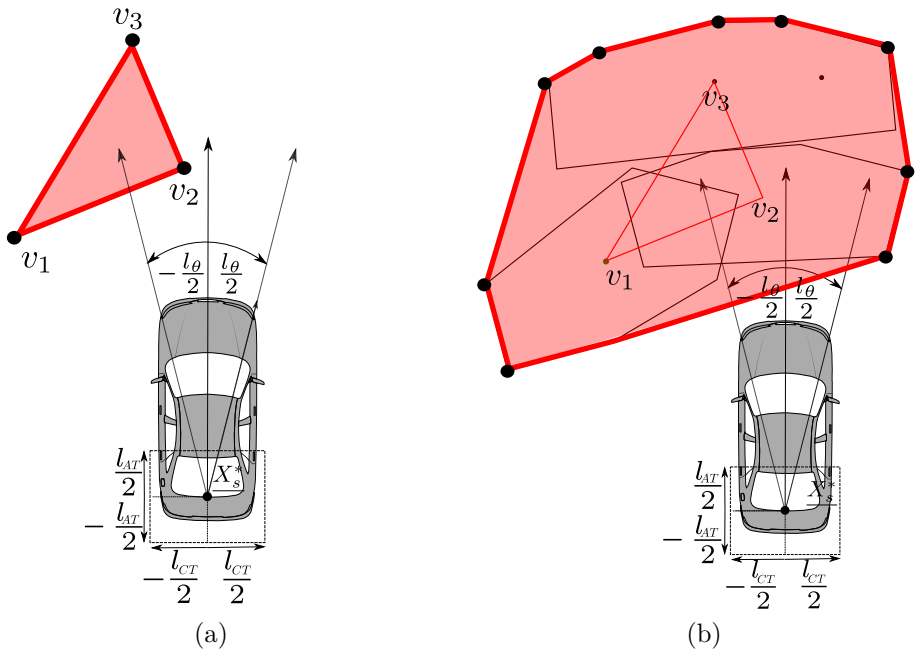


Figure 4.8: Illustration of the extended convex hulls computation of the confidence domain of a perceived object. In, Figure 4.8a a perceived object is bounded by a polygon with three vertexes. The confidence domains the vehicle pose are also depicted. In Figure 4.8b, the extended convex hull is built from the confidence domain of each vertex. It represents the confidence domain of the perceived object.

As all the computations are performed in the vehicle reference frame, the transformation f is directly applied on the vertexes of the previously computed confidence

domain to get the final occupancy domain in the world frame. Unlike the previously presented approach, no approximation is necessary to propagate such confidence domain. In the following, this method will be called “direct method”.

4.4.5 Clusters Filtering with the HD Map

Once the 2D occupancy of the detected objects has been computed with the localization uncertainty, the HD map can be used to filter out all the objects that are not on the drivable space. A cluster is not considered on the road surface if all the vertexes of its bounding polygon lie outside of the road and no edge intersects any road boundary. Conversely, a cluster is considered as lying on the road if all the vertexes of its bounding polygon is within the boundaries of the road surface, and no edge intersects any road boundary. The rest of the clusters are ambiguous and are considered as uncertain. The resulting classification is therefore constituted by three categories: “road”, “not road” and “uncertain”.

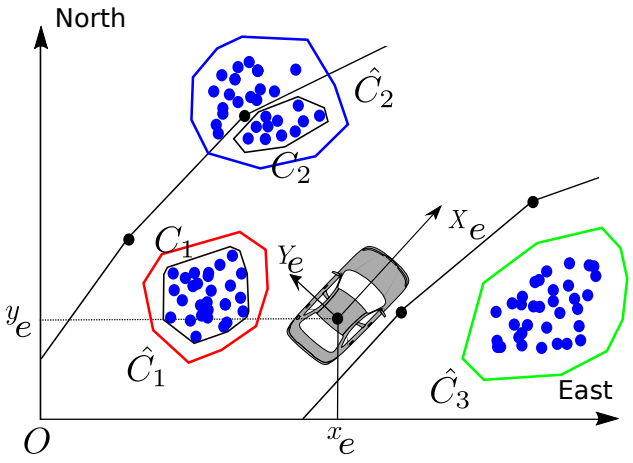


Figure 4.9: Extended convex hulls computed taking into account the AD vehicle localization uncertainty. Three classes are present: not-road (green), road (red), uncertain (blue). The black convex hulls represent the hulls computed by taking into account the AD vehicle ground truth localization, highlighting the differences between the two cases.

Figure 4.10 depicts the result of this classification on a driving scenario that involves some MD vehicles in a roundabout.

4.4.6 Lane-level Curvilinear Occupancy

So far we have seen how to transfer the AD vehicle uncertainty into the perception result in order to obtain a consistent environment perception. Once the 2D space occupied by a perceived object has been computed, it is possible to use once more the information contained in the HD map to compute the object occupancy at the lane level. Indeed, the interactions between the AD vehicle and the surrounding road users can be performed at lane level rather than at Euclidean space level. This concept has already been discussed in detail in chapter 3. Therefore, the curvilinear occupancy intervals generated by each obstacle along the driving lanes is more informative than the 2D occupancy space. In the available HD map, in

Figure 4.10: Experimental result of the HD map filtering applied to extended bounding polygons. The green hulls represent the obstacles outside the region of interest, while the red hulls represent vehicles inside it. The blue hulls represent objects partially contained in the region of interest. Notice that the resulting bounding polygons include the AD vehicle uncertainty.

addition to the road borders, the middle of each lane of the road is encoded as a polyline, *i.e.*, a sequence of line segments.

To compute the curvilinear occupancy of a 2D bounding polygon, first the intersection between such a polygon and the polygons obtained from the representation of each road lane is computed. The obtained result is a sub-polygon representing the 2D space effectively occupied on such lane. This occupancy can occur on multiple lanes, depending on the positioning of the bounding polygon.

After this step, each vertex of this sub-polygon is map-matched onto the HD map lanes, according to the lanelet method explained in section 7.3.3. To make the computation faster, the HD map has been organized with a graph-based structure and a breadth-first search algorithm is used during the map-matching step.

The aforementioned operation is performed on all the obstacles being classified as "road" or "uncertain", in order to keep the integrity constraint. In fact, not considering an object which confidence domain intersects a portion of the drivable surface, even if small, could lead to collisions. On the other hand, some heuristics can be proposed to handle objects that can lead to false positive during the occupancy computation (e.g. bushes or trees at the road sides).

Regarding the polygon intersection with the HD map link borders, two different situations may occur:

1. *Only one lane is intersected*: The resulting map matching operation is performed directly on the original occupancy polygon vertices. A single occupancy interval is defined by the minimum and the maximum curvilinear abscissa of all the points on the polyline.

2. *Several lanes are intersected*: This may occur when a vehicle is changing lane or when the occupancy space is too large. This situation implies that there is an ambiguity regarding which lane results being effectively occupied. From an

integrity point of view, every candidate lane has to be considered as occupied by a curvilinear interval.

The final output of this step is a list of occupancy intervals that represent the space occupied by all MD vehicles along the HD map polylines (see Figure 4.11).

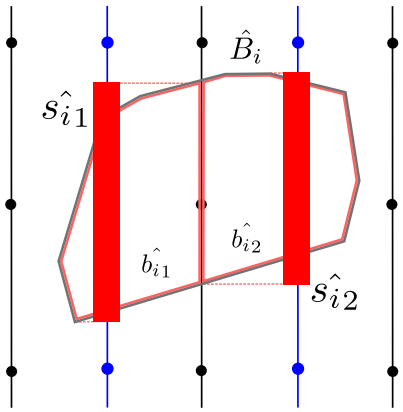
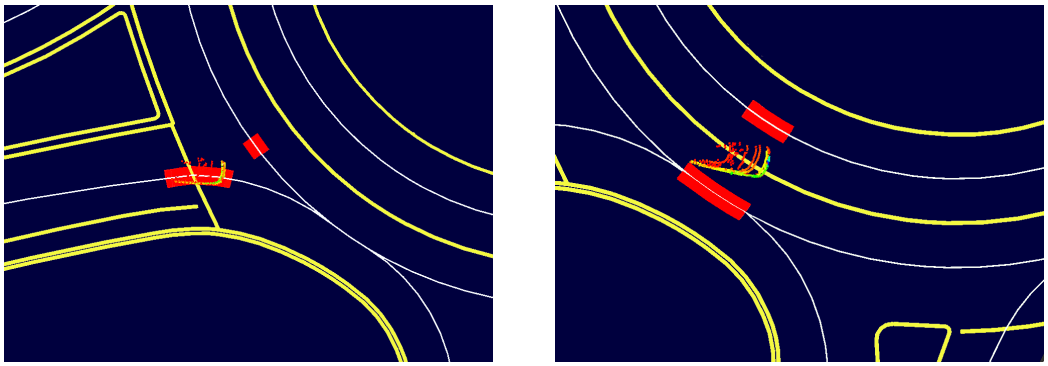


Figure 4.11: The red polygon $\hat{B}_i$ intersects two different lanes generating two sub-polygons $\hat{b_{i1}}$ and $\hat{b_{i2}}$. All the vertices of each sub-polygons are map-matched on the polyline in the middle of the occupied lane. The result is a set of curvilinear intervals representing lane level occupied space $\hat{S}_i = \{\hat{s_{i1}}, \hat{s_{i2}}\}$.

Figure 4.12 typical situations where multi-map matching can occur in practice.



(a) (b)

Figure 4.12: Results of the lane-level curvilinear occupancy extraction from the 2D bounding boxes. Figure 4.12a: occupancy intervals generated by a vehicle entering into a roundabout. Figure 4.12b: occupancy intervals from a vehicle driving between two lanes. For both figures, occupancy is computed along several lanes.

4.5 Integrity Definition of Perceived Objects

The method proposed in the previous section has been designed with a particular attention to the integrity of the spatial occupancy of the perceived objects. Such a definition means that the perception system must be able to provide information with a certain degree of confidence and in a robust way.

In this case, the integrity of the perceived environment needs to follow two statements:

- It must be consistent w.r.t. the real object occupancy.
- It must be robust to localization uncertainty.

In the first case, it is required that the system provides the exact occupancy of every detected vehicle. In particular, the system must avoid both missed detection and wrong occupancy detection. Such statement implies that the bounding polygons of the perceived objects must be representative as much as possible of the true object occupancy. Nevertheless, this depends on the perceived information and it is not always trivial to determine the whole object occupancy from a partial or incomplete LiDAR-based representation. Indeed, it is not always possible to clearly understand the object planar dimensions and if the computed bounding polygon corresponds to the real object occupancy. As a consequence, an incomplete detection of the whole occupancy of a vehicle can lead to severe consequences in terms of navigation safety, as for example in the case of inter-distance keeping and vehicle following. To cope with that, in the next chapter a tracking system with the help of an infrastructure-based technique to compensate such lack of perception has been proposed.

Concerning the second statement, the system has to handle the localization uncertainty directly into the perception layer. This means that, if one injects localization uncertainty into the result of the perception system, the perception should remain robust to localization errors. Given an estimation of an unknown quantity and a risk α, it is possible to build a confidence domain that contains this unknown quantity with probability equal to $1 - \alpha$. A confidence domain is said consistent if the aforementioned probability is greater or equal to $1 - \alpha$. In this case, we consider that the integrity property is respected.

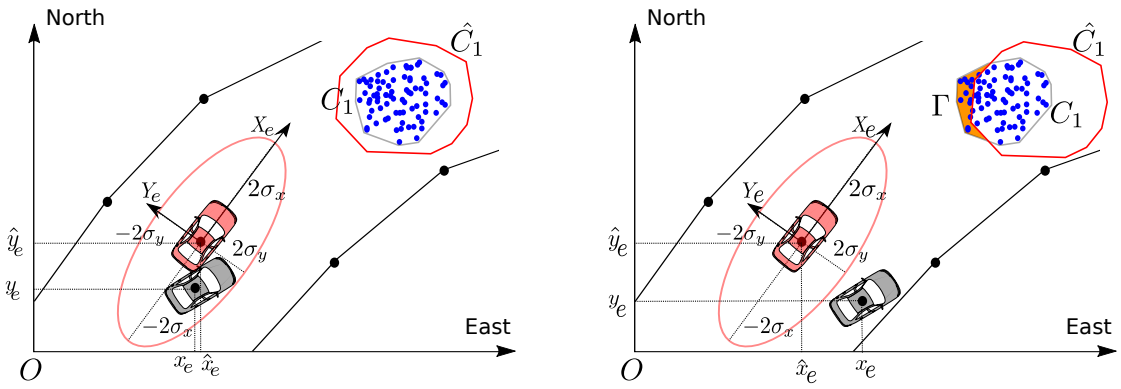
In order to apply the definition to the case of bounding polygons, an extension of what was previously said for points to clusters has been done. Considering that, if the extended convex hull $\hat{C}_i$ contains all the LiDAR points of the cluster that originates $\hat{C}_i$ with a probability of $1 - \alpha$, it means that the integrity property is respected. In other words, in $(1 - \alpha) \cdot 100$ percent of the cases, $\hat{C}_i$ includes all the perceived LiDAR points belonging to the originating cluster. This definition holds true in the case where the ground truth localization ${}^{w}X_e$ of the AD vehicle is contained in the ellipse built from the observed localization ${}^{w}\hat{X}_e$.

Figure 4.13 depicts the reasoning. Figure 4.13a shows the case in which the ground truth localization ${}^{w}X_e$ is contained in the ellipsoid centered on the observed AD vehicle pose ${}^{w}\hat{X}_e$ and computed exploiting its the covariance matrix Σ_e.

In this case, the resulting extended convex polygon $\hat{C}_1$ contains entirely the convex polygon C_1 built considering the ground truth localization (which is, of course, unknown in real-life applications). Figure 4.13b illustrates the dual case where the ground truth localization is not contained in the ellipsoid generated by the observed pose ${}^{w}\hat{X}_e$. Consequently, the resulting extended convex polygon $\hat{C}_1$ is not guaranteed to contain all the LiDAR points of the originating cluster. As it is shown, in such case, some unsafe regions Γ arise.

Based on this reasoning, we define the integrity of a bounding polygon w.r.t. its originating cluster as follows.

Definition: Given a raw-data cluster G_i, composed of LiDAR points g_i, and a transformation function $f(\cdot)$ that converts LiDAR points g_i from the LiDAR sensor frame to the world frame. By applying $f(\cdot)$ to the bounding polygon B_i computed with the localization ground truth ${}^{w}X_e$, one obtains a polygon that bounds points of G_i, and a risk α. A bounding polygon $\hat{B}_i$, obtained after having applied the transformation $f(\cdot)$ with the estimated localization ${}^{w}\hat{X}_e$, satisfies the integrity property



(a) (b)

Figure 4.13: Integrity extended to clusters bounding polygons: If the true pose is contained in the ellipsoid of uncertainty centered on the observed pose, then the extended hull fully contains the points (Figure 4.13a). If not, some points are outside (Figure 4.13b).

if and only if:

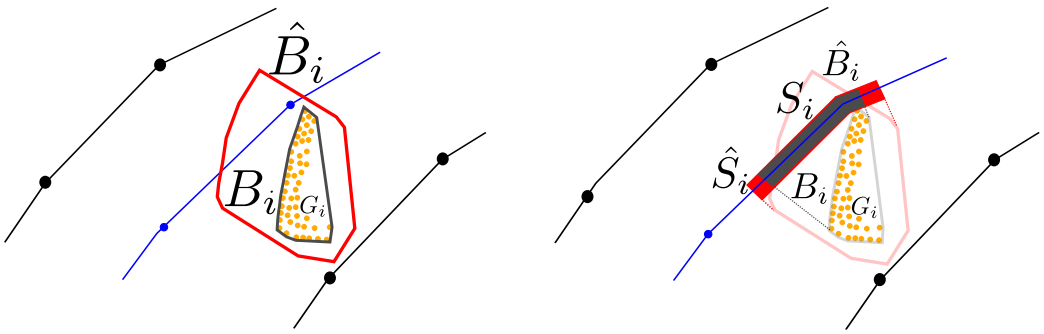
$$Pr(B_i \subset \hat{B}_i) \geq 1 - \alpha \tag{4.9}$$

with $\hat{B}_i = f\left(g_i, {}^w\hat{X}_e\right)$ and $B_i = f\left(g_i, {}^wX_e\right)$.

These concepts can also be formulated at the lane-level curvilinear occupancy in a similar fashion by considering the occupancy intervals instead of the bounding polygons. Given a set of segments S_i representing the lane level drivable space occupied by a bounding polygon B_i and a risk α, a set of segments $\hat{S}_i$, computed by map matching the transformed bounding polygon $\hat{B}_i$, satisfies the integrity property if and only if:

$$Pr(S_i \subseteq \hat{S}_i) \geq 1 - a. \tag{4.10}$$

These integrity definitions are depicted in Figure 4.14.



(a) 2D occupancy integrity (b) Lane level integrity

Figure 4.14: Integrity representation in terms of 2D occupied space (a) and curvilinear occupied space (b). The red hulls and segments take into account localization uncertainty while grey ones are obtained from the ground truth.

4.6 Results

In this section, results obtained to validate the perception pipeline are presented and discussed. These results have been generated thanks to an experimental platform to record datasets in roundabouts. The data contains real traffic data collected on some public roundabouts in the cities of Compiègne and Rambouillet. The whole detection pipeline has been coded in C++ with the middleware ROS. The experimental validation has been carried out both on recorded data and during real tests with experimental vehicles.

4.6.1 Map Classification and Filtering with Uncertainty

In order to test the behavior of the HD map-based filtering based on clusters convex hulls computed with the localization uncertainty, a comparison between the HD map-based filtering with uncertain convex hulls and the HD map-based filtering obtained with the corresponding ground truth has been performed. The aim of such tests is to check how the HD map-based filtering changes when the uncertainty on the AD vehicle localization grows.

To obtain the bounding boxes computed with the localization ground truth instead of the uncertain one, the same pipeline has been used. To compute the extended bounding polygons, the set-membership propagation method (Section 4.4.3) has been used to inject the uncertainty in the bounding polygons. Once one has obtained both the hulls computed using the ground truth and the extended ones, it is possible to apply the HD map filter to classify their relative position w.r.t. the region of interest. This classification step is repeated for both hulls computed using the ground truth and the extended ones. Then, a comparison between the two is carried out. For a given bounding polygon, one is interested in checking its classification result in both the aforementioned cases. Table 4.1 reports the overall percentage of classification transitions for every bounding polygon when uncertainty in AD vehicle localization is added. As one can see, when we switch to the case with the localization uncertainty the class “uncertain” has been added.

Observing Table 4.1, it is possible to see for each cluster how the classification result has changed when uncertainty has been added (in other words, how the uncertainty has changed the relative positioning of an MD vehicle w.r.t. the HD map). As previously said, some dangerous situation may arise when adding uncertainty. Such situations are false positive and false negative cases. The first one corresponds to a case where, due to localization uncertainty, an object previously classified as a not-road point becomes road. The second case corresponds to a point that switches from road to not-road.

Both cases must be avoided. However, false negatives are more dangerous than false positives. If one observes again Table 4.1, it is possible to see that neither false negative nor false positive cases occur. This means that the method does not provide misleading information to the AD vehicle. It is also possible to observe that no phantom objects appear when we switch to uncertain representation. Such a point provides an improvement in system availability in the context of AD vehicles navigation.

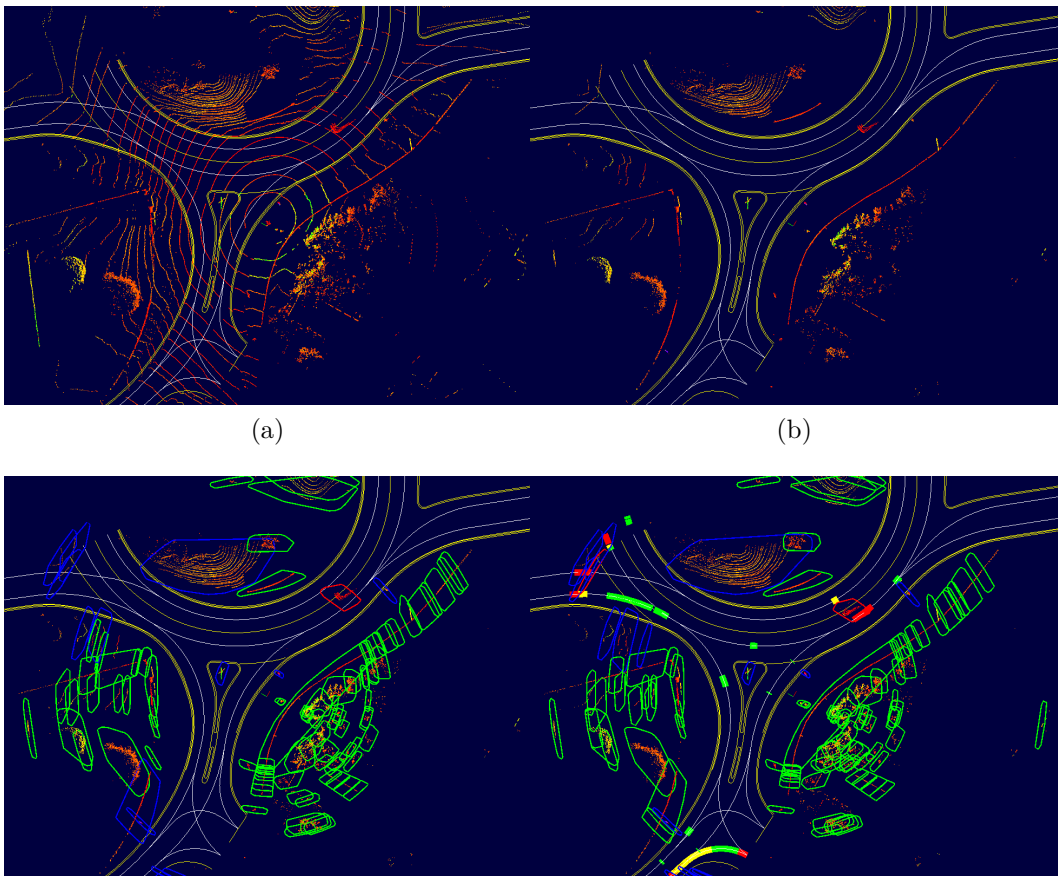
However, when uncertainty is added, a certain percentage of the objects switch from road or not-road to uncertain. This is because when bounding polygons tend to grow due to the uncertainty injection, they have more probability to intersect a lane border. This phenomenon is more likely for road objects because they are closer

to lane borders than uncertain ones, which can be distributed everywhere outside the road surface.

Table 4.1: Classification results when localization uncertainty is added (percentage). Notice that neither False Positive nor False Negative cases occur.

Class Changing			
Ground Truth\Uncertain	Road	Not road	Uncertainty
Road	33.26	0	66.754
Not road	0	91.84	8.16

Finally, a qualitative overall result that illustrates the extended convex hull classification based on the AD vehicle uncertain positioning can be seen in Figure 4.10.



(a) (b)

(c) (d)

Figure 4.15: Overall result of the MD vehicles detection pipeline in the case where AD vehicle uncertain localization is present. Figure 4.15a: the raw data LiDAR point cloud. Figure 4.15b: non-ground points extraction. Figure 4.15c: extended bounding polygons computation. Figure 4.15d: curvilinear occupancy extraction.

The whole detection pipeline with the AD vehicle uncertain localization propagation is computed in around 33.965 ms on a laptop with a 2.2 GHz CPU, which gives an average working rate of around 29 Hz. Such a rate is significantly below the threshold of 100 ms that is required to run a system in an AD vehicle navigation framework. Moreover, the whole algorithm running time is around 11.567 milliseconds for traffic scenarios that can contain up to 200 clusters. The computation times

of each block for the case where AD vehicle localization is treated are summarized hereafter in Table 4.2.

Table 4.2: Average Computational Time (ms)

Uncertain Computational Time					
Step	Segmentation	Clustering	Map Filtering	Map Matching	TOT
Time	7.743	9.317	11.567	5.075	33.702

4.6.2 Integrity Analysis: Simulation Study

In this section, a comparison of the performance between the set-membership propagation method explained in section 4.4.3 and the state-of-the-art linearization method described in section 4.4.2 is carried out. The aim of the test is to check whether the integrity property is preserved when the localization uncertainty is propagated to perception results via the two methods.

To test and compare the two methods, a case study in a simulated environment has been implemented. First, it generate a random point which represents a simulated single LiDAR point cluster. Then, we compute the bounding polygon that surrounds it using the ground truth pose of the AD vehicle. After that, an uncertain vehicle localization is generated from a Gaussian distribution centered on the ground truth.

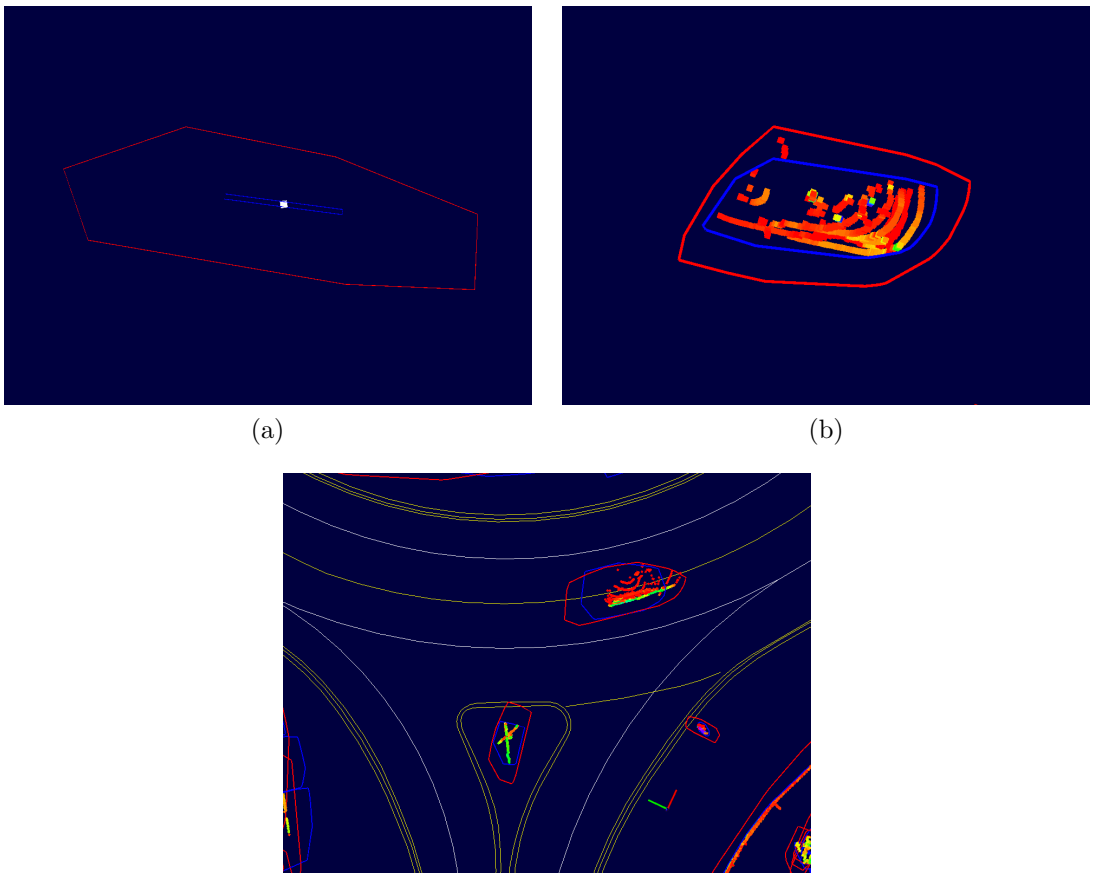
As the transformation between the ground truth pose and the uncertain one is perfectly known, we can apply the same transformation to the LiDAR point, in order to simulate the acquisition of such point from a sensor situated in a position different from the ground truth one. In this situation, a new extended bounding polygon is computed taking into account the uncertain AD vehicle localization. Once this computation has been done, we check if the new extended bounding polygon $\hat{C}_i$ contains entirely the ground truth one C_i according to the integrity definition given in section 4.5. The scenario of this test is depicted in Figure 4.16a.

The same reasoning is extended to a LiDAR cluster that represents a static object. We used an experimental car as an obstacle and we have extracted its representation from the LiDAR point clouds data. Figure 4.16b illustrates this process. Again, we perform the same test that we did for the previous case.

Table 4.3: Integrity results for the three use-cases described in section 4.16 for a risk value of $\alpha = 0.05$.

Overall Integrity Ratio		
Dataset	Set-membership propagation	Linearized propagation
Single point	~ 1	0.95
Static object	0.978	0.867

Table 4.3 reports the overall percentage of integrity for each method and for each scenario for an integrity risk $\alpha = 0.05$. The set-membership method outperforms clearly the one based on linearization. Indeed, the performance of the linearized propagation is below the expected confidence value for the second scenario, and so it does not reach the integrity objective.



(a) (b)

(c)

Figure 4.16: Comparison results for the three different use-cases. Figure 4.16a illustrates the case with a single point. Figure 4.16b illustrates the case of a static cluster. The blue polygon represents the linearized propagation bounding polygon, while the red polygon represents the one obtained with the set-membership propagation. Figure 4.16c depicts the same reasoning for a scene contained in the datasets.

4.6.3 Integrity Analysis: Real Tests

To evaluate the performance of the method in both Cartesian and curvilinear coordinates, several real experiments have been carried out. A dataset has been recorded at the entrance of a roundabout in the cities of Compiègne and Rambouillet, France. This dataset is composed of ten sequences of about 10 minutes of road traffic, with a moderately dense traffic flow. Moreover, the Renault ZOE experimental vehicle was equipped with a Velodyne VLP-32 LiDAR and a GNSS/IMU NovAtel SPAN-CPT with PPK corrections to provide a ground truth localization. To simulate errors on the localization, a Gaussian noise was injected into the ground truth pose provided by the localization sensor. The standard deviation values used for such noise are $\sigma_x = 0.1$ m, $\sigma_y = 0.16$ m and $\sigma_\theta = 0.01$ rad.

First, the 2D occupancy of the linearized and set-membership method from an integrity point of view has been computed. The integrity property refers to the capability of the estimated occupation space, computed from an uncertain localized sensor, to contain the whole cluster of point cloud or bounding polygon representing the object perceived at their exact localization.

Table 4.4: Average computational time in ms for the whole processing (segmentation, clustering, uncertainty propagation, map filtering and map matching).

Step	Segm.	Clust.	Propagation	Map Filt.	Map Match.
set-membership	7.753	9.317	0.183	11.384	5.075

Considering a risk α and the confidence level $1-\alpha \in \{0.90, 0.95, 0.99, 0.999, 0.9999\}$, the occupancy confidence domain has been computed by using both methods. Then, for each obstacle cluster, it has been computed whether all the points within the cluster, acquired from the exact localization, were contained in the occupancy confidence domain or not. A method satisfies the integrity constraint if the ratio of clusters entirely contained in the confidence domain is greater than the confidence level $1-\alpha$. Table 4.5 summarizes the results obtained in the experiments. It can be seen that the linearized method does not provide reliable results. This is due to the non-linearity of the heading error propagation that leads to a *banana* shaped distribution of the LiDAR points which is badly approximated by a Gaussian. On the other hand, the set-membership approach keeps the integrity up to a confidence of 99%. For higher degrees of confidence, the integrity objective is not exactly reached but the statistics remain very close.

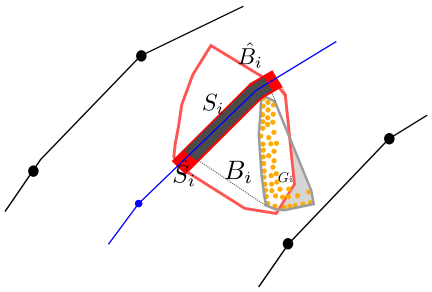


Figure 4.17: As the red polygon $\hat{B}_i$ does not contain the gray cluster B_i entirely, it does not satisfy the 2D integrity property. Its projection $\hat{S}_i$ at lane level along the blue polyline, however, includes the true curvilinear occupancy S_i.

The same analysis has also been performed in terms of occupied curvilinear space. Table 4.6 shows that the set-membership propagation method also outperforms the

Table 4.5: Integrity comparison of the linearized and set-membership methods for 2D occupancy confidence domain.

$1-\alpha$	90%	95%	99%	99.9%	99.99%
Linearized	44.54%	49.50%	56.45%	62.59%	66.33%
set-membership	97.69%	98.87%	99.21%	99.69%	99.88%

Table 4.6: Integrity comparison of the linearized and set-membership methods for lane-level occupancy confidence domain.

$1-\alpha$	90%	95%	99%	99.9%	99.99%
Linearized	91.04%	91.50%	91.96%	92.82%	93.05%
Set-membership	99.23%	99.30%	99.35%	99.73%	99.90%

linearization one at lane level. This is directly related to the better performance identified in 2D space occupation. Nevertheless, the curvilinear integrity values are significantly higher than the 2D occupation ones. This behavior is due to the fact that a 2D bound that does not fully include a given obstacle may still include its curvilinear occupancy at the lane level. Figure 4.17 illustrates such a situation.

4.7 Conclusion

In this chapter, we have shown a method that provides a robust and consistent perception of the surrounding driving environment for AD vehicle navigation. This method is based on the combination of Velodyne LiDAR data and a HD map to extract the most relevant objects on the road surface. A full pipeline to process the points acquired by a Velodyne LiDAR has been developed. This processing flow also uses the data provided by a HD map, making the proposed work an innovative way to perform perception for AD vehicles.

The case of uncertain vehicle localization has been taken into account by developing a new method that transfers it directly on LiDAR points in a set-membership manner. The proposed approach has a good level of integrity and works better than a method that uses linearization to propagate the uncertainty. This can be explained by the fact that, as objects are far away from the LiDAR sensor, uncertainty on the heading angle has a great impact. A linearization method provides only an approximation of such uncertainty. The set-membership propagation method propagates the whole uncertainty to bounding polygons rather than approximating it. For this reason, its bounding polygons remain consistent in all the use-cases.

In order to evaluate the performance experimentally, we have used a ground truth localization provided by an accurate GNSS receiver with PPK corrections. We found out that the performance is very good in terms of integrity. Moreover, the whole pipeline is processed in quite a low computational time, performing more than twice faster than the minimum safety requirement for autonomous navigation, proving that the proposed approach is viable for real-time autonomous vehicle navigation in safety-critical scenarios. This has been demonstrated at the IEEE Intelligent Vehicles symposium in Paris in June 2019 [66]. Autonomous driving tests were also conducted on the Séville experimental test track of the University of technology of Compiègne. A vehicle and a bicycle were looping inside a roundabout with the aim of having the autonomous vehicle inserting in the roundabout. These tests with a 95% confidence level led to satisfactory behavior for autonomous navigation.

5 Cooperative Road Users Tracking with Intelligent Infrastructure

5.1 Introduction

This chapter presents a cooperative perception system able to provide consistent and meaningful information to the AD vehicle about the surrounding traffic environment. This approach is studied with the goal to ensure safe AD vehicle navigation in complex urban scenarios. In chapter 3, we have presented a decision-making strategy for complex driving maneuvers using a HD map while taking into account the uncertainties. In chapter 4, we discussed an on-board LiDAR perception system that detects and localizes the road users with a special attention to the propagation of the uncertainty on localization since this strategy relies on working in a fixed working frame in which the map is expressed.

In order to obtain a safe navigation by using the algorithm described in chapter 3, it is required to get a consistent curvilinear occupancy of the perceived objects and a good knowledge of their speeds along the polylines of the map. Indeed, the system must be able to estimate correctly the entire occupancy of perceived objects several seconds ahead in order to guarantee a safe navigation during the roundabout crossing process.

To achieve this goal we propose to implement a tracking system that first exploits the information provided by on-board LiDAR perception to estimate the occupancy and the speed of the other vehicles. We are particularly interested in investigating if this system provides information that meets the integrity criteria. In a second stage, it will be studied how the information from a remote intelligent infrastructure perception system can extend the field of view and improve the integrity of the perceived objects in order to robustify the AD vehicle on-board perception system. This idea comes from cooperative perception systems. An example of this work can be found in [7, 86] and [33]. In the particular case of a data-fusion approach with an intelligent infrastructure, another example can be found at [62, 88]. Furthermore, other details about cooperative and collaborative systems state-of-the-art have been detailed in Chapter 2.2.1.

This chapter is organized as follows. After having reviewed classical tracking systems, Section 5.3 will describe the method that have been used in this system to track detected objects from LiDAR bounding polygons with the help of the HD map. Section 5.4 will present the main aspect of the remote intelligent infrastructure perception system. In section 5.5, the main details about the techniques that have been adopted to implement a cooperative data fusion strategy between the intelligent infrastructure and the AD vehicle perception system are presented. Moreover, a comparison between the obtained results with the ones obtained exploiting only the AD vehicle information is reported too. Finally, in section 5.6.5 a discussion about the experimental results is provided where we highlight the main advantages and drawbacks of the use of an intelligent and communicating infrastructure for enhancing safety of AD vehicles navigation.

5.2 Tracking Framework

5.2.1 State-of-the-Art

Moving object tracking is widely used in several domains, as for example traffic monitoring, surveillance and aircraft detection. In the context of AD vehicles navigation, the main purpose of tracking methods is to estimate values related to surrounding vehicles and to estimate quantities that are not directly observable from the provided raw data. An example of this is the speed and acceleration of other vehicles that are not directly observable from LiDAR point clouds. Moreover, tracking-based methods can also be used to provide a more consistent and smooth estimation of the evolution of state variables over time. Finally, it is also possible to recursively estimate the degree of uncertainty on the estimated state variables along with their estimation process. In the case of the simultaneous tracking of several objects, a data association algorithm is necessary to perform a matching between the detected observations and the existing tracks to do a tracking update or to create new ones.

The main role of a tracking system is to estimate the states of external dynamic objects over time. Every object is represented by a set of state variables that are considered necessary and sufficient to describe its behavior and to perform the navigation task successfully. Considering the existing algorithms on this topic in the literature, the Kalman Filter is recognized by the majority of researches as the most common and widely used technique for this purpose [77]. The execution flow of such a filter is divided into two main steps. First, there is a prediction step where the estimation of a track state is propagated to the present time using only an evolution model. Then, in the estimation step, the predicted state is fused with the information coming from the measurements and an update of the state is done. Some others examples of tracking techniques can be found at [29, 31, 33, 61]

In order to run a Kalman filter, it is required to define an evolution and an observation model for modeling the system. Based on the different models used for both the state and the observations, different types of Kalman Filter can be identified. For a both linear evolution and observation models, the Kalman Filter (KF) can be applied. In the case where observation and/or evolution models are non-linear, an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF) is classically used. In order to obtain a good performance during the estimation process, it is required that measurement noises and uncertainty in the evolution model follow a white zero mean Gaussian distribution. If not, the KF is not guaranteed to converge towards the optimal solution during the estimation process.

To cope with that, some other techniques can be used instead. In particular, advanced Bayesian filters can replace a KF. This approach has the advantage that it does not require the noise to follow a Gaussian distribution. For instance, it is suited in the case of multi-modal noise distributions. The most common implementation of this filter is the particle filter. Such a filter is based on a Monte Carlo method for generating a set of particles that approximate the system state posterior distribution. Each particle has a weight representing the probability of being an estimate of the system state. The filter alternates propagation and re sampling steps, assigning at every time new weights to the particles according to the updated information computed with the received observations. The set of the particles with their corresponding weights corresponds to an approximation of the probability distribution of the system state. Despite the fact that this algorithm requires less hypotheses on the system modeling, it demands a significant amount of computation power to be implemented in real-time without some additional dedicated hardware.

In the case of multi objects simultaneous estimation, a data association procedure is required to understand which observation corresponds to which track. For multiple objects tracking, a set of Kalman filters can be run in parallel and each single filter performs the estimation of the state of a single object. In order to correctly associate observed data to the correct filter, a data association step needs to be solved. In this context, an estimation of the state of a single object is called a "track". At each iteration of the filtering process, the data association performs the following operations. Firstly, it associates tracks to observations. Then, it creates a new track if an observation has not been associated to an existing track and it removes a track if such a track has not been assigned to any observations for a certain amount of time. Considering the main state-of-the-art algorithms, we can summarize the main techniques used to perform this task in the following list:

- Nearest Neighbor (NN): Assignment of each track to its closest observation according to a certain distance. Such a distance can be the Euclidean or the Mahalanobis distance.
- Global Nearest Neighbor (GNN): It works as the NN approach. However, assignment is done by globally minimizing the distances between tracks and observations, ensuring that all the tracks are assigned to an observation. To compute the assignment, some heuristics as the Hungarian algorithm can be used.
- Joint Probabilistic Data Association Filter (JPDAF): The probabilities of each association hypothesis considering all the possible assignment combinations are computed. After that, the probability of associating a track i to an observation j is calculated by the sum of all the probabilities that satisfy the assignment. The final assignment is done by validating the most probable one.
- Multi-Hypothesis Tracking (MHT): Instead of using the hypothesis only for the association probability computation as it is done in JPDAF, a subset of such probabilities is propagated to update a track with multiple observations and then create multiple possible tracks of the same object.

The presented tracking and data association algorithms can be combined to perform data association and tracking according to the considered scenarios. Some approaches that illustrate the deployment of such techniques in the cases of LiDAR based objects tracking are [60, 100]. Moreover, tracking algorithms can be used to track not only vehicles but also other road users as well. Some examples of tracking algorithms for pedestrians can be found in [44, 87].

5.2.2 Kalman-Based Moving Objects Tracking

An important stage of a tracking system is the filtering strategy based on a state space modeling. In the literature, there exists a wide range of approaches to perform this task. In this work, we have decided to use the Extended Kalman Filter (EKF). The EKF has been chosen because it provides a good compromise between the complexity of the algorithm and our specific needs.

5.2.2.1 State Modeling

An EKF is a state estimation technique suited for non-linear dynamic systems. In its most general form, a non-linear dynamic system (with no input because we do

tracking of external and uncontrolled objects) can be represented by the following model:

$$\begin{cases} X_{k+1} = f(X_k) + \alpha_k \\ Y_k = g(X_k) + \beta_k \end{cases} \tag{5.1}$$

Equation 5.1 describe both the evolution and observation models which contain non-linear equations. The vector X_k indicates the system state at a given time step k, Y_k is the observation vector, α_k is the model noise that represents uncertainty about the system model, and β_k is the measurement noise. The main working hypothesis of the Kalman Filter is that both the noises α_k and β_k are assumed to be represented by white Gaussian random processes having zero mean. We have therefore that $E(\alpha_k) = E(\beta_k) = 0$. Furthermore, covariance matrices of these random processes defined as $Var(\alpha_k) = Q$ and $Var(\beta_k) = R$ have to be known. Finally, we assume that the two noises α_k and β_k are not correlated.

In this work, we use the notation $\hat{X}_{k|k}$ to represent the estimated track state at time k, while the notation $\hat{X}_{k|k-1}$ represents the predicted state at time k based on the estimate $\hat{X}_{k-1|k-1}$ computed at time $k-1$.

In tracking methods, once a new track is created, it has to be correctly initialized in order to begin the estimation process. Once the initialization of all the aforementioned quantities is performed, the EKF consists of two subsequent phases: the prediction step, used to predict the value of the state up to the time of the measurements, and the estimation step, where the state value is updated according to the new measurements.

To make the overall estimation process converge to a consistent state estimation, it is required to perform a tuning of the noise covariance matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{\mathrm{m} \times \mathrm{m}}$, where n is the dimension of the state vector and m is the dimension of the measurements vector. This can be done using recorded data and ground truth. Often, the state covariance matrix $P \in \mathbb{R}^{\mathrm{n} \times \mathrm{n}}$ of the estimation error is initialized with a diagonal matrix that contains weight coefficients accordingly to each state variable.

5.2.2.2 Prediction

In order to run the EKF, the Jacobian matrix of the evolution model has to be computed. It is needed to linearize the non-linear model around the state estimate $\hat{X}_{k-1|k-1}$. This matrix is computed as follows:

$$A_k = \left[\frac{\partial f}{\partial X} (\hat{X}_{k-1|k-1}) \right] \tag{5.2}$$

In the prediction step, the state of a track is predicted by using the evolution model $\hat{X}_{k|k-1} = f(\hat{X}_{k-1|k-1})$, extrapolating the state estimate at time $k-1$ up to time k. In a similar way, the prediction of the state covariance matrix $P_{k|k-1}$ (of the prediction error) at time k is obtained as:

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q \tag{5.3}$$

5.2.2.3 Estimation

As previously said, the aim of this steps is to integrate the acquired measurements to update the state prediction $\hat{X}_{k|k-1}$ obtained at the previous step. To do so, first

an observation Y_k is elaborated at time k from the raw exteroceptive measurement. Let us first define a matrix C_k in the following way:

$$C_k = \left[\frac{\partial g}{\partial X}(\hat{X}_{k-1|k-1}) \right]$$

The Kalman gain K is then calculated as:

$$K_k = P_{k|k-1} C_k^T (C_k P_{k|k-1} C_k^T + R)^{-1} \tag{5.4}$$

After that, the prediction of the state is updated in the following manner:

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k (Y_k - C_k \hat{X}_{k|k-1}) \tag{5.5}$$

The covariance matrix of the estimation error can be updated using Joseph's form:

$$P_{k|k} = (I - K_k C_k) P_{k|k-1} (I - K_k C_k) + K_k R K_k^T \tag{5.6}$$

In practice, the tracking system often encounters situations where multiple detections occur at the same time. This happens if there is more than one vehicle in the driving scenario at the same time. For this reason, before performing the estimation step, a data association algorithm is executed to assign each incoming observation to the most likely track. An additional step is performed to handle not updated tracks and not associated observations. This aspect is detailed in the next paragraph.

5.2.3 Data Association

To find a correspondence between an existing track and an observation, the assignment of tracks to detections is based on the maximization of a certain metric. First, a cost matrix $M_{N_t \times N_o}$ is created after the prediction step. The dimensions N_t and N_o correspond to the total number of tracks and observations. Each element of the cost matrix corresponds to a value that expresses the likelihood between a track and an observation. In the literature, there exists several methods to compute such a metric [46]. In this work, we use the Mahalanobis distance. For a given track j and an observation i, the corresponding squared Mahalanobis distance is computed as:

$$D_{ij}^2 = \left(Y_i - g\left(\hat{X}_{k|k-1}\right)_j\right)^T \left(C_k \left(P_{k|k-1}^{-1}\right)_j C_k^T + R_i\right)^{-1} \left(Y_i - g\left(\hat{X}_{k|k-1}\right)_j\right) \tag{5.7}$$

where the notation is the same as the one used before for the EKF description.

Once the cost matrix has been computed, the assignment of the detections to the tracks is performed by using the Hungarian algorithm [52]. This heuristic attempts to globally minimize the tracks to observation association cost expressed by the metric defined in Equation 3.9. The resulting pairing is given as input to the EKF estimation step.

5.2.4 Tracks Management

The aforementioned technique allows to perform data association between existing tracks and observations in order to associate them and to carry on the tracking process. However, to complete this step, it is necessary to define a policy for handling the creation and elimination of tracks. In particular, it is necessary to create a new

track when an observation is not associated to an existing track, in order to include this new information in the list of tracked objects.

Conversely, the deletion of a track is performed when a track is not associated to an observation for a given time period. This process allows to remove tracks that represent objects that are no more on the AD vehicle field of view. To model the overall elapsed time to decide when a track can be eliminated, often a variable is used to count the number of consecutive iterations when a track was not associated to any observation. A threshold on this parameter is set to choose when eliminating tracks. This counter allows the tracker to keep alive tracks that do not receive observation a few time instants because of short occlusions for instance.

The last process of the track management is the merging of close tracks in order to fuse together tracks that represent the same object. This provides a more compact representation of the tracked objects which can be of importance when there are numerous objects to track. However, in our approach, it is crucial to keep in memory a representation as complete as possible of the dynamic objects in order to fulfill the integrity criteria. Moreover, the number of tracks remains small in practice. So, we have decided not to perform the merging step between tracks.

5.3 Map-aided Road Objects Tracking using On-Board LiDAR

5.3.1 Problem Statement

In this section, a tracking system capable to localize the road users in the world frame using a 3D LiDAR installed high up on the roof of the vehicle is presented. To do so, one needs to have a precise knowledge of the localization of the AD vehicle (position and heading). As previously seen, the navigation algorithm needs objects occupancy and speed along the polylines. The snapshot raw LiDAR clusters are not an adequate information for several reasons. First, speed is missing and second, lane occupancy can be badly estimated in some situations. A tracking system is an interesting solution to get a more robust and consistent information about the moving objects and to fulfill the integrity criteria discussed before. A tracking system is also a mean to estimate the derivatives of the position of a tracked object and consequently the speed vector.

Moreover, in order to improve the estimation of vehicles states and to provide an occupancy according to the curvilinear formalism, it has been decided to exploit again the information contained into the HD map at several levels. We will see in this section how the HD map information is introduced to implement a map-aided tracking algorithm.

The choice of a map-aided tracker is motivated by the following points that resume the main goals of this stage:

- To estimate as best as possible the occupancy of the other vehicles (real or virtual) on the AD vehicle path,
- To estimate as best as possible the speeds of the other vehicles (real or virtual) along the polylines.

In order to fulfill the aforementioned statements, two approaches for developing the tracking system have been considered: the curvilinear formalism and the Cartesian one.

A curvilinear tracker is a tracking system that is entirely based on tracking objects along the polylines of the HD map. In this tracking method, the evolution model is computed in curvilinear coordinates and the observation consists in the projection of the measurements along the HD map polylines. This method has the main advantage that it does not involve an explicit computation of the heading angle, which is a well known challenging task for LiDAR-based perception systems. Heading is not necessary because the evolution of the tracks is constrained along the polylines of the map.

However, this method has several drawbacks. Firstly, the evolution model in curvilinear coordinates can be tricky to handle (in particular during the transition from one polyline to another). Secondly, as tracked objects are constrained to move only along polylines, the algorithm can lead to poor performance in situations where tracked vehicles are far from the HD map polylines, as for example in a curve road or during a lane-change maneuver. This is, in general, due to the fact that HD map polylines represent the center of the lanes, which is not always the nominal path of vehicles during navigation. Finally, it has been observed experimentally that the distance along the polylines is often higher than the distance traveled by the vehicle and, for this reason, the tracker estimation is often delayed which requests to add an important evolution noise [14].

Conversely, if one considers a Cartesian 2D tracker, tracked objects are free to move in the 2D space. In this case, a good estimation of the heading angle is necessary to perform an acceptable prediction of the state. In order to properly update the heading information, an accurate and consistent measurement of the heading must be done from the observations on the perceived objects. This task is very challenging in the case of LiDAR-based perception [31]. This is because the heading estimation is based on the shape of the point cloud that represents the perceived object. In general, this shape depends on the orientation of the object w.r.t. the AD vehicle. For this reason, it is challenging to provide an accurate heading estimation using a LiDAR. Moreover, the estimation of the heading uncertainty (which is necessary to implement a tracker) is also a not trivial task to achieve.

A compromise between the two aforementioned approaches can be found by using a map-aided tracking system. This solution allows to implement a Cartesian tracker without using a measurement of the heading angle in the tracking process. Instead of that, the heading will be inferred from the HD map polylines, allowing a more robust solution than the 2D Cartesian case.

Furthermore, a map-aided tracker allows to feed the tracks with observations projected onto the map polylines (in Cartesian coordinates). This is helps the tracker to converge more quickly towards the HD map polylines, letting the tracks the freedom to move away from them when tracked objects do not follow the road center. This approach also needs to use a good covariance matrix relative to the matched point along the polyline, in order to take into account the uncertainty of the observation in the tracking process. Section 5.3.3 presents the proposed solution to this problem.

Finally, this approach also allows to instantiate several tracks that correspond to the same object in order to model all the possible MD vehicle unknown behaviors. This will be explained in section 5.3.6.

To achieve the aforementioned objectives, it has been decided to develop a tracking system able to track both the pose with speed and the length of detected vehicles. For each vehicle, the tracked variables are composed by the position x and y, its heading angle θ and its speed v. Furthermore, it has also been tracked the vehicle

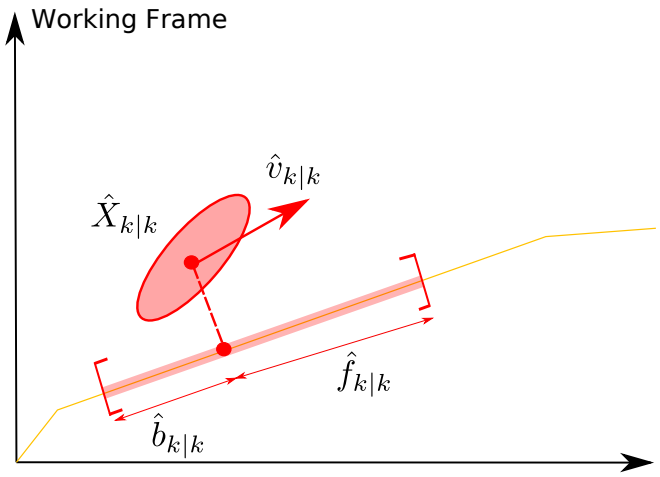


Figure 5.1: Track representation. The information about the position, speed and heading is contained into the vector $\hat{X}_{k|k}$ and its uncertainty $P_{k|k}$, while the length along the map is represented by the red occupancy interval along polylines. Variables $\hat{b}_{k|k}$ and $\hat{f}_{k|k}$ are used to track the length. The estimated speed vector $\hat{v}_{k|k}$ is not necessarily parallel to the map polyline.

length along the longitudinal motion direction (i.e. the polylines of HD map). For this task, two variables f and b have been introduced to estimate the half-lengths behind and ahead the location of the centered point of the detected vehicle. The occupancy on the polylines of HD map is computed at the end, when the tracker provides an estimate.

5.3.2 Tracks Representation

Let us see how we represent a single track instance in the tracking method. As previously said, this tracking system exploits the HD map information to better encompass the occupancy of detected objects at lane-level. Moreover, a separation between the kinematic state of a track and its estimated length projected on the polylines is done. These two pieces of information will give the occupancy necessary for navigation. For the first part, the information contained in the estimate of the state vector $\hat{X}_{k|k}$ is represented in Cartesian coordinates and it is free to move in the 2D space, instead of being constrained only to HD map polylines. The same reasoning can be applied to the state covariance matrix estimate $P_{k|k}$. Regarding the second part, the length of a detected track is estimated along the HD map polylines. For every track instance, a curvilinear interval along the polylines is therefore estimated. To track the size of the interval, it has been decided to introduce in the tracking process two variables $\hat{b}_{k|k}$ and $\hat{f}_{k|k}$. Figure 5.1 illustrates this representation for a given track.

5.3.3 Map Observation Uncertainty Computation

The aim of this part is to choose a method to provide a good representation of the uncertainty for the map observations used by the tracker.

Let us come back to the method of the previous chapter to build objects bounding polygons of the LiDAR clusters. Consider the polygon of one LiDAR cluster without propagating the AD vehicle localization uncertainty (in blue in Figure 5.2). Its barycenter is the LiDAR measurement Z_k. This measurement is map-matched to give the observation $Y_k = \begin{bmatrix} x_k, & y_k \end{bmatrix}$.

Now, we are looking for a way to compute the uncertainty of this observation thanks to the curvilinear coordinates. In particular, it is required to have a covariance matrix in the world reference frame. To do that, first the covariance is computed (denoted ${}^{L}R$) along the polyline of the HD map.

Considering the integrity definition, for a given risk α, one can obtain a $1-\alpha$ confidence interval by using the following formula:

$$Pr(\hat{X} - \gamma\sigma \leq X \leq \hat{X} + \gamma\sigma) = 1 - \alpha \tag{5.8}$$

Given this interval, one can propagate uncertainty to point Z_k accordingly to section 4.4.3. That being done, a new convex polygon is obtained. Such a new polygon represents the uncertainty on the barycenter of the cluster (in green in Figure 5.2). Exploiting again the method proposed in section 4.3, it is possible to project the obtained bounding polygon onto the HD map polylines and extract its occupancy on both along and cross directions. Finally, considering a given risk $\alpha = 0.05$, we have $\gamma = \Phi^{-1}(0.025) \simeq 2$. With this value one can also compute the variance along a polyline by now using as input a confidence interval obtained from the occupancy projection according to the following equation:

$${}^{L}\sigma_s = \frac{|s_{k,max} - s_{k,min}|}{4} \tag{5.9}$$

where $s_{k,min}$ and $s_{k,max}$ are the minimum and maximum abscissa of all the edges of the polygon along the polyline.

The same reasoning can be done also for the transverse component n of the curvilinear pose. Figure 5.2 depicts this computation for a given barycenter of a convex hull. This leads to a covariance matrix ${}^{L}R$ in the form:

$${}^{L}R = \begin{bmatrix} {}^{L}\sigma_s^2 & 0 \\ 0 & {}^{L}\sigma_n{}^2 \end{bmatrix} \tag{5.10}$$

This matrix is computed in the frame L of the polyline. As one can see, this matrix is diagonal in this frame. It means that, during this computation, the hypothesis that the along error is uncorrelated with the cross error has been made. Therefore, the obtained covariance matrix is oriented along the main axis of the local HD map frame. In order to obtain the covariance matrix R in the world reference frame (it is the matrix R of the observation of the tracker), the angle ψ of the polyline has been considered to perform a rotation according to the following:

$$R = \mathrm{Rot}(\psi).{}^{L}R.\left(\mathrm{Rot}(\psi)\right)^{T} \tag{5.11}$$

Where the transformation $\mathrm{Rot}(\psi)$ corresponds to a 2D rotation matrix computed for a given angle ψ. Here the hypothesis that the angle ψ has a negligible uncertainty has been made too.

Equation 5.11 provides a matrix in the form:

$$R = \begin{bmatrix} \sigma_x^2 & \sigma_{x,y} \\ \sigma_{x,y} & \sigma^2{}_x \end{bmatrix} \tag{5.12}$$

Where term $\sigma_{x,y}$ represents correlations that appear between the two variables x and y when passing from the polyline frame L to world frame w. Once this step is done, the obtained matrix R represents the uncertainty on the object barycenter Y map-matched along the polyline map in the world reference frame. This quantity is used to handle the uncertainty about a given observation. Moreover, it is also

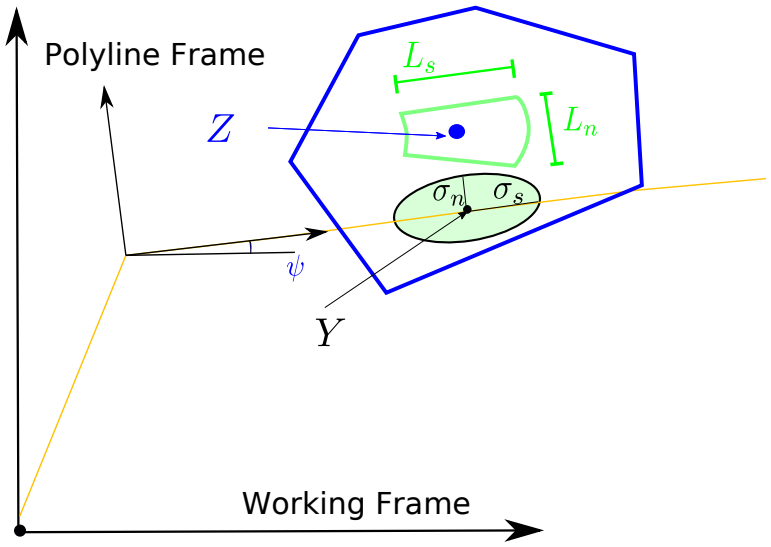


Figure 5.2: Map-aided observation Y and its uncertainty. The blue polygon is the bounding LiDAR cluster. The LiDAR measurement Z is the barycenter of this polygon. The map-aided observation Y is the projection of point Z on the polyline. The green polygon represents the direct propagation of the localization uncertainty on point Z and the green ellipsoid represents the resulting uncertainty on Y with the standard deviations σ_s and σ_n in the polyline frame.

used to perform data association and consequently fused with an existing track in order to update its state. This method has the advantage of generating a covariance matrix in its general form and exploits the heading information provided by the HD map.

5.3.4 Map-Aided Tracking of the Pose

In order to obtain a relevant tracking information for the navigation strategy, we propose to add a soft constraint coming from the map to make the motion of a detected object attracted to the polylines of the HD map. This allows to obtain a representation of the dynamic objects that matches well with the map-based navigation framework. Moreover, the tracking is implemented in the world reference frame, rather than the sensor's frame. It is an allocentric approach that relies on a good localization of the AD vehicle.

The estimated state of each track is defined as $\hat{X}_k = \begin{bmatrix} \hat{x}_k & \hat{y}_k & \hat{\theta}_k & \hat{v}_k \end{bmatrix}^T$ where $\hat{x}_k, \hat{y}_k$ are the track coordinates in the world reference frame, $\hat{\theta}_k$ the heading in this frame and $\hat{v}_k$ the speed along the heading direction. For the sake of simplicity, the superscript index ${}^w(\cdot)$, that indicates the world frame has been omitted. The point $(\hat{x}_k, \hat{y}_k)$ represents the barycenter of the detected vehicle. In this work, the LiDAR observation corresponds to this point. This observation is used for the track update.

Let us define a LiDAR polygon measurement as Z_k. We suppose that this point corresponds to the center of an object. Z_k is transformed into an observation Y_k by performing a map-matching of Z_k along the HD map polylines. The resulting map-matched point is $Y_k = \begin{bmatrix} x_k & , y_k \end{bmatrix}^T$.

Here, a tracker using only one evolution model has been considerd. For road users, a constant velocity and constant yaw rate model has shown to be a good choice [85]. The non-linear discrete-time evolution model for a given track $\hat{X}_k$ is

therefore defined as:

$$X_{k+1} = f(X_k) = \begin{bmatrix} x_k + \triangle_t v_k \cos(\theta_k) \\ y_k + \triangle_t v_k \sin(\theta_k) \\ \theta_k \\ v_k \end{bmatrix} + \alpha_k \tag{5.13}$$

where $\triangle_t$ is the elapsed time since the last iteration. In practice, $\triangle_t$ is not really constant in this system because it depends on the frequency of the LiDAR data. The term α_k represents the model noise and Q is its covariance matrix.

$$Q = \triangle_t \begin{bmatrix} q_x & 0 & 0 & 0 \\ 0 & q_y & 0 & 0 \\ 0 & 0 & q_\theta & 0 \\ 0 & 0 & 0 & q_v \end{bmatrix} \tag{5.14}$$

$\{q_x\, q_y\, q_\theta\, q_v\}$ are tuning parameters that represent the power of the model noise. In particular, q_x and q_y are about the uncertainty on the position propagation, q_θ is about the heading variation and q_v is for the speed change over time.

The Jacobian matrix A_k of the prediction step is as follows:

$$A_k = \left[\frac{\partial f}{\partial X}(\hat{X}_{k-1|k-1})\right] = \begin{bmatrix} 1 & 0 & -\triangle_t \hat{v}_{k-1|k-1} \sin(\hat{\theta}_{k-1|k-1}) & \triangle_t \cos(\hat{\theta}_{k-1|k-1}) \\ 0 & 1 & \triangle_t \hat{v}_{k-1|k-1} \cos(\hat{\theta}_{k-1|k-1}) & \triangle_t \sin(\hat{\theta}_{k-1|k-1}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.15}$$

The tracks are initialized as follows. When a track is initialized by duplicating another one, it retrieves the attributes of its mother track. When it is a new track because of a new object detection, we proceed as follows:

$$\begin{cases} \widehat{x}_{k-1|k-1} = {}^w x_k \\ \widehat{y}_{k-1|k-1=} {}^w y_k \\ \widehat{\theta}_{k-1|k-1} =^w \psi_k \\ \widehat{v}_{k-1|k-1} = v_n \end{cases} \tag{5.16}$$

v_n is the average speed of the lane of the object. This information can be observed experimentally and registered in the map. From a Bayesian point of view, this is the best prior that one can have.

A good heading initialization is of prime importance. It is obtained from the corresponding angle of the lanelet frame ψ_k. For a given detected object, the curvilinear representation of the observation is represented as follows (the superscript on the left indicates that the coordinates are in lanelet):

$${}^L Y_k = \begin{bmatrix} s_k \\ n_k \\ \psi_k \end{bmatrix} \tag{5.17}$$

Where the superscript index L indicates that the coordinates are curvilinear along the HD map polylines. As previously said in section 4.3, it has been assumed that a conversion between the point ${}^L Y_k$ and Y_k always exists. Moreover, the initial values of the position correspond to the x and y coordinates of the observation Y_k.

Let us now focus on updating the tracks with LiDAR measurements. After several tests, it has been decided that this step can be broken down into two phases: position and heading.

When the update occurs, the tracking algorithm has performed a propagation of the previous state estimate up to the time instant of the received measurements according to Equation 5.1. If the heading is correctly estimated, the state has been approximately propagated along the tangent vector of a polyline. Please note that in this tracking strategy, we make the assumption that the observations are provided at a sufficiently high rate, in order to avoid a significant drift of the model since the prediction is performed along the heading vector given by $\hat{\theta}_k$. To associate the observations (a LiDAR scan provides in general several clusters) with their corresponding tracks, the Hungarian data association procedure with Mahalanobis distances has been used (see section 5.2.3). For every good match, the predicted state $\hat{X}_{k|k-1}$ is updated into $\hat{X}_{k|k}$.

The position observation model is linear and defined as follows:

$$Y_k = CX_k + \beta_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \end{bmatrix} + \beta_k \tag{5.18}$$

Regarding this update stage, no linearization is therefore required and the map-matched observation Y_k along the HD map polylines acts as a soft constraint in the tracker. In the update step of the EKF, the covariance matrix of this observation has been presented in section 5.3.3.

The map can also provide knowledge on the heading of the track, which is very useful to improve tracking. However, this should be done only if the track is close to a polyline to enable a good tracking when a dynamic object changes lanes. As a result, we proceed as follows. If the angular deviation between the track heading and the polyline angle is less than a chosen threshold (5 degrees for example), then we use the following observation model:

$$Y_{\theta,k} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \end{bmatrix} + \beta_{\theta,k} \tag{5.19}$$

where $Y_{\theta,k}$ is the heading of the polyline at the map-matched point. The variance of the noise $\beta_{\theta,k}$ is a tuning parameter which can be easily adjusted because it has a very concrete physical meaning. In this specific case, a value of $var(\beta_{\theta,k}) = (5 \cdot \frac{\pi}{180})^2 rad^2$ has been chosen.

5.3.5 Length Estimation of the Tracked Objects

The exploitation of a HD map-based formalism is useful not only for improving the estimation of the pose and speed of the tracks but also to compute at the lane-level the occupancy of the tracked vehicles. For this, it has been proposed to include in the tracking system the information about the length of the tracks projected on to the polylines. The main objective is to get a reliable estimation of the object occupancy. This is crucial to avoid potential collisions when the estimated lane occupancy is given as input to the navigation methods described in chapter 3.

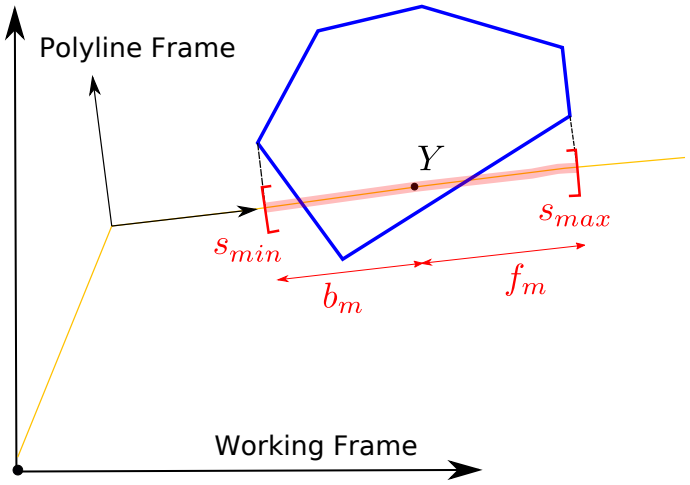


Figure 5.3: Observation of the length of a detected object. It is the hull of the bounding polygon of the LiDAR cluster in blue projected on the map. The blue cluster is the same as the one of Figure 5.2. b_m and f_m are respectively the measured backward and forward lengths w.r.t. point Y.

Let us consider the occupancy curvilinear interval $[s_{k,min}, s_{k,max}]$ associated to the curvilinear occupancy of map-matched vertices of an observed bounding polygon (without the inclusion of the localization uncertainty of the AD vehicle). Such an interval is computed along the HD map polylines and reflects the portion of the lane occupied by a detected matched object (see Figure 5.3).

A simple approach to well estimate the length of track is to include in the state of the tracker two new variables b and f that keep track of the backward and forward lengths that describe the curvilinear occupancy from the estimated center of the track.

A question that might arises is the following: how one can define an evolution model for propagating the lengths during the prediction step? The solution that has been proposed is to keep constant the lengths during the prediction step and to perform a "max update" based on the comparison of the estimated lengths and the corresponding observed lengths along the map-matched polyline and to keep in memory the largest one, as it is shown hereafter:

$$\hat{b}_{k|k} = max(\hat{b}_{k-1|k-1}, b_{m,k}) \tag{5.20}$$

$$\hat{f}_{k|k} = max(\hat{f}_{k-1|k-1}, f_{m,k}) \tag{5.21}$$

in which $b_{m,k}$ and $f_{m,k}$ represent backward and forward lengths of the measured curvilinear occupancy interval relative to the Y_k observation.

In general, $f_{m,k} \neq b_{m,k}$ because in the case of bounding polygons coming from LiDAR measurements, the position of Y_k depends on the shape of the convex hull. Furthermore, Y_k can be different from the barycenter of a detected object.

There are different ways to update the length variables. If one knows (or supposes) that Y_k represents the barycenter of a detected object, then, he can carry out the computation as follows. Let L_k be the observed length of the vehicle, i.e. $L_k = s_{k,max} - s_{k,min}$. In such a case, one can write:

$$\hat{b}_{k|k} = max\left(\hat{b}_{k|k-1}, \frac{L_k}{2}\right) \tag{5.22}$$

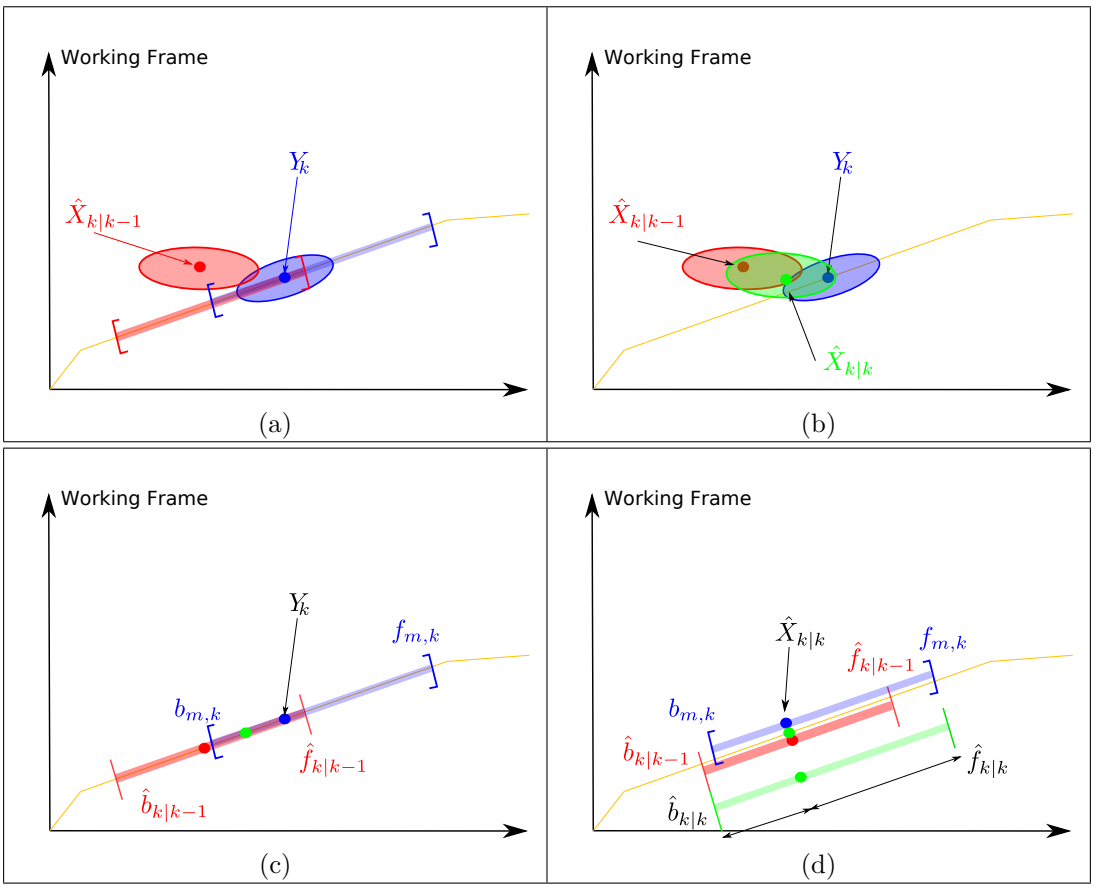


Figure 5.4: Update process of a track. 5.4a illustrates the matching of a track prediction (red) with an observation (blue). 5.4b depicts the update process of the pose only. 5.4c shows only the length intervals of the prediction (red) and the observation (blue). 5.4d describes the length update process (green).

$$\hat{f}_{k|k} = max\left(\hat{f}_{k|k-1}, \frac{L_k}{2}\right) \tag{5.23}$$

Now, if one has the knowledge that Y_k represents the front of the car, one can write $s_{k,max} = 0$ and $s_{k,min} = L_k$ and vice-versa in the dual case.

This approach works as follows. First, estimate $\hat{X}_{k|k}$ and the observation Y_k are aligned and the maximum between the predicted occupancy $\hat{f}_{k|k-1}$, $\hat{b}_{k|k-1}$ are computed. Then, their corresponding observations $f_{m,k}$ and $b_{m,k}$. Figure 5.4 depicts the interval information associated to a given observation.

The "max update" approach prevents the estimated occupancy to shrink and it updates the current state only if a bigger occupancy has been estimated. The occupancy interval of a track can only increase over time which is pessimistic but reliable in terms of integrity. This is coherent with the idea of avoiding misleading information, even if it can happen that objects occupancy are overly-estimated. One possible solution to cope with this issue could be to perform the lengths updates according to some criteria, for example by introducing an algorithm that identifies outliers in order to prevent an excessive growth of our intervals.

5.3.6 Multi-hypothesis objects Lane Occupancy

Another interesting aspect about a HD map-aided tracking algorithm is that it can handle multi-hypothesis occupancy for a detected object during the tracking

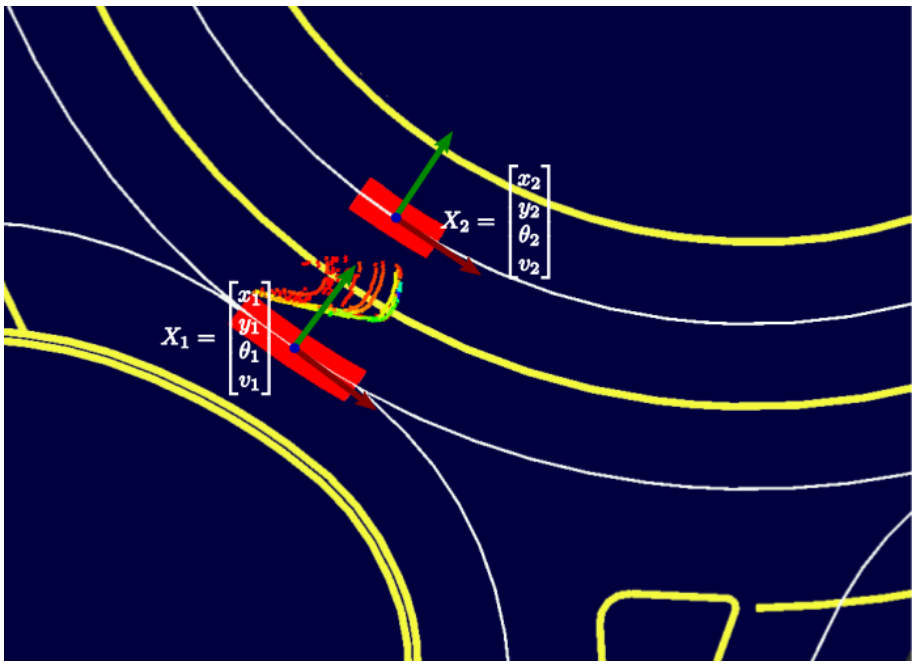


Figure 5.5: An example of multi-hypothesis tracking for a single detected object. Notice that the two tracks X_1 and X_2 are treated independently w.r.t. each others.

stage by using a multi-hypothesis occupancy criteria. This allows to create several instances of tracked objects in order to model the occupancy of tracked objects at lane-level. This idea finds a perfect match with the multi-hypothesis criteria explained in sections 3.2.3 and 3.3.3 for handling uncertain driving behavior at the decision step. Now, it allows to handle the uncertainty about vehicles unknown locations.

In this part of the work, the tracking is made to handle several instances of the same MD vehicle. This means that, when there are more than one interval occupancy obtained from the same object, the tracker keeps track of all the possible hypotheses individually, in order to encompass all the possible MD vehicle behaviors.

To illustrate this concept, Figure 5.5 gives a case where a multi-hypothesis object tracking occurs. It can be seen that this behavior typically appears when a tracked object is moving from a center-lane to another. In the case of the figure, the MD vehicle is leaving the outermost lane to perform a lane change maneuver. For this reason, lane occupancy appears slightly on both lanes. Notice that, when the vehicle finishes its lane change maneuver, there won't be any observation on the outermost lane that matches to it and this first hypothesis will be removed.

As a consequence, it may happen that the system has more tracks w.r.t. the actual number of MD vehicles present in the driving scenario. In this work, we have decided to treat the estimation process of each track instance originated from the same object independently from the others. In other words, we let each track evolve independently from the other tracks. Nonetheless, for each track instance that represents a given detected object, we attribute a label that represents this relationship. This can be useful to implement a more optimized criterion for handling tracks association, creation and elimination. The creation and elimination of the tracks are implemented immediately after the data association step. Further details about this particular stage and the strategy adopted to handle tracks creation and deletion will be given in the next section.

5.3.7 Integrity-based Data Association and Tracks Handling

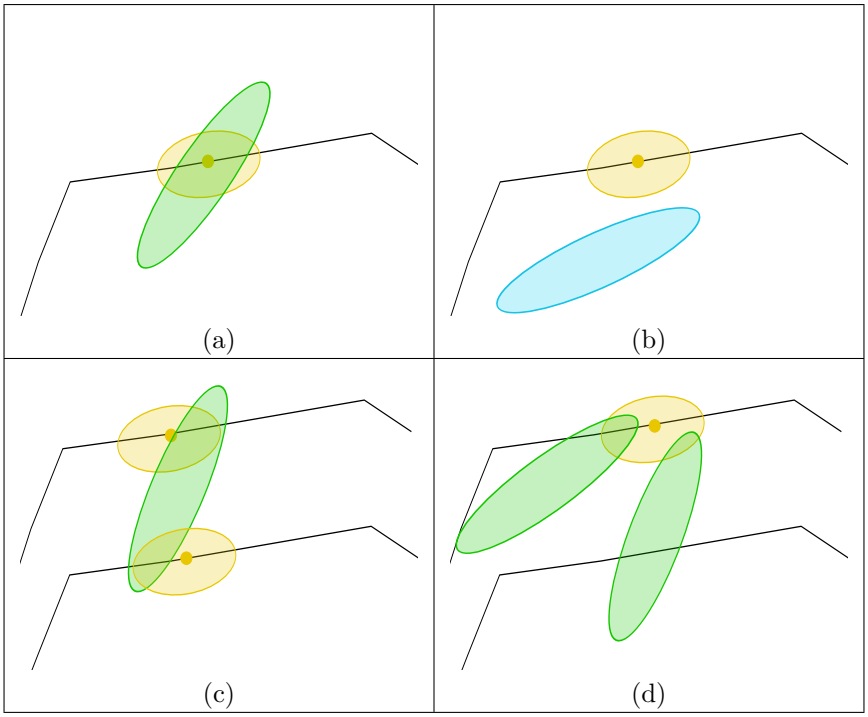


Figure 5.6: Several cases that can occur during the data association phase. The orange point represents the map-matched observation and its uncertainty and the orange intervals represent its occupancy at lane-level. The ellipses represent existing tracks (in blue when there is no association) .

To get a high integrity system during the tracking stage, we propose in this section to discuss about different data association strategies. A general overview about the data association technique that have been used in this work has been already provided in section 5.2.3, where the association criterion and the heuristic used to assign observations to existing tracks has been discussed. Nonetheless, the focus here is on improving the aforementioned strategy to achieve a data association that provides high integrity of the perceived objects in this particular case.

To get a high level of integrity, it has been considered that every detection must be associated to a track. This is done in order to avoid missed detection of the perceived objects. In other words, if N_o observations and N_t existing tracks are present, the inequality $N_t \geq N_o$ must hold at every step of the tracking stage. This means that, during the data association step, either one observation is associated to an already existing track, or a new track is created to track the new detection. This approach can be seen as redundant on the one hand, however, on the other hand, it allows to represent every single object detection without losing information about perceived objects.

Figure 5.6 illustrates several situations that can occur during the data association step and how the tracker handles these cases in order to provide the maximum level of integrity. As one can see from Figure 5.6a, when there is an unique association between an existing track and an observation, the observation is merged with that track. Conversely, when the observation is too far away from the existing track, as in the case of Figure 5.6b, a new track is created to incorporate the new information in the tracked objects. In the case where a track that is associated to multiple observations, as in the case of Figure 5.6c, the observation that maximizes the

association criterion will be chosen. Regarding the unassigned observation, they will be used to initialize a new track to include the new observation into the tracked objects.

Finally, when multiple tracks that can be associated to a single observation are present, as in the case of Figure 5.6d, the track that better satisfies the association criterion will be chosen. Regarding the other track, it will be left unassociated and its number of skipped frames will be incremented. If, for a given track, the number of consecutive skipped frames is greater than a fixed threshold, the track is deleted.

In order to maintain a more compact representation of the tracks, instead of counting the number of skipped frames to remove tracks from the list of tracked objects, one can exploit the duplication mechanism seen in section 5.3.6 to keep track of duplicated tracks that have been created from a parent track. This technique allows to know how many different tracks relative to a single object are present. In order to satisfy the integrity condition, one can remove tracks immediately after when a track has not been associated for the first time. However, before doing that, one needs to check if at least one instance of the track has been associated during this step. If it is the case, the track is kept alive and only the unassigned instance is removed. Conversely, the whole set of track instances is removed.

5.3.8 Lane Occupancy Estimation

In the final step of the tracking stage, it is required to provide an estimate of the detected vehicles position, speed and occupancy to be sent to the decision-making level. As previously seen, information that is necessary to this step is elaborated in several phases. The EKF estimates objects position and speed and the forward and backward lengths are estimated in parallel with a max strategy as the perceived shape of a moving object changes over time when using a LiDAR sensor.

Accordingly to the integrity objective, the aim is to get an object occupancy that can be as much as possible consistent with reality. For this reason, the estimated length of a tracked object occupancy is enlarged with the estimated uncertainty about the tracked object position.

Let us consider a track estimate $\hat{X}_{k|k}$ in the world reference frame. It has a certain uncertainty represented by the covariance matrix $P_{k|k}$. This uncertainty needs to be included in the final estimation of the occupancy interval.

In practice, the estimates $\hat{b}_{k|k}$ and $\hat{f}_{k|k}$ are taken and a confidence interval is added to them:

$$s_{max} = s + \hat{b}_{k|k} + 2\sigma_l \tag{5.24}$$

$$s_{max} = s - \hat{f}_{k|k} - 2\sigma_l \tag{5.25}$$

where the term σ_l represents the uncertainty on the longitudinal direction of the matrix $P_{k|k}$ in the polyline frame. This quantity is obtained by projecting the ellipse dimensions along the road path frame via the scalar product. This can be interpreted in the following way: an uncertainty on the object position is propagated to the final estimate of the object occupancy. As a consequence, the final estimate encompasses this uncertainty to provide a reliable estimation of the occupied space. Notice that if the object position is perfectly known, i.e. $\sigma_l = 0$, Equation 5.24 and 5.25 provide the same results as Equation 5.20 and Equation 5.21. Figure 5.7 gives an illustration of this process.

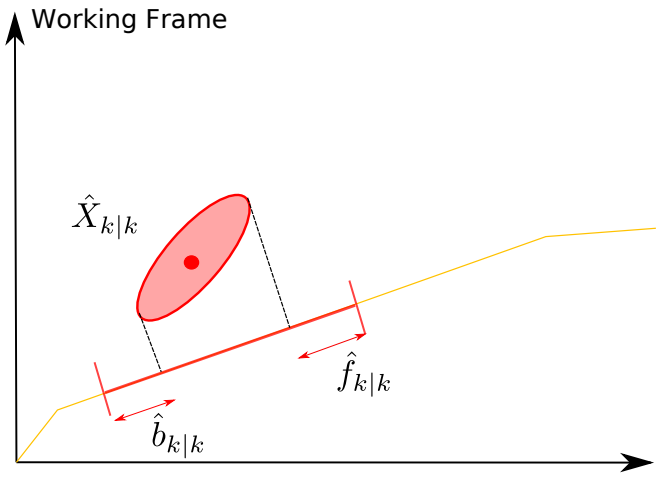


Figure 5.7: The final lane-level occupancy computed as the sum of the occupancy and the localization uncertainty of the estimated track.

5.4 Remote Infrastructure Perception System

In this section, we describe the main components of the intelligent infrastructure system that has been considered in this work. This system has been used for detecting MD vehicles and to provide additional information about them to the AD vehicle in real-time. This system has been developed in the framework of the Tornado project by the Université Gustave Eiffel.

5.4.1 System Overview

The goal of this system is to exploit a remote intelligent infrastructure to provide additional information about a given complex urban scenario. Combined with the AD vehicle on-board sensors, it can provide enhanced and redundant perception information of the ongoing driving situation. In order to associate the data from these two sources, a data fusion approach needs to be performed.

The focus here is on the problem formulation and the description of the main image-processing steps. In the Tornado project, it was decided to install two cameras to monitor the most dangerous branches of a roundabout on the left side of the AD vehicle entering branch. This choice has been done in order to provide the AD vehicle a sufficient field of view for the decision-making and navigation algorithms. Figure 5.8 illustrates the positions where the two intelligent cameras were installed. One can observe that, for every AD vehicle trajectory of the roundabout crossing use case, these cameras provide additional information w.r.t. the left side of the roundabout and its entering branches.

Furthermore, each camera has been installed on a road pole and its position referenced into the HD map. This is helpful for knowing the exact location of the intelligent infrastructure and the projection of its perception results in a HD map-based framework. Figure 5.9 depicts the whole system installed. This intelligent system allows to detect the vehicles on some zones of the roundabout that the on-board perception is not able to see.

5.4.2 Images Processing

Images processing has been performed by the researchers of the Université Gustave Eiffel. To detect the vehicles in the acquired infrastructure images, the detection

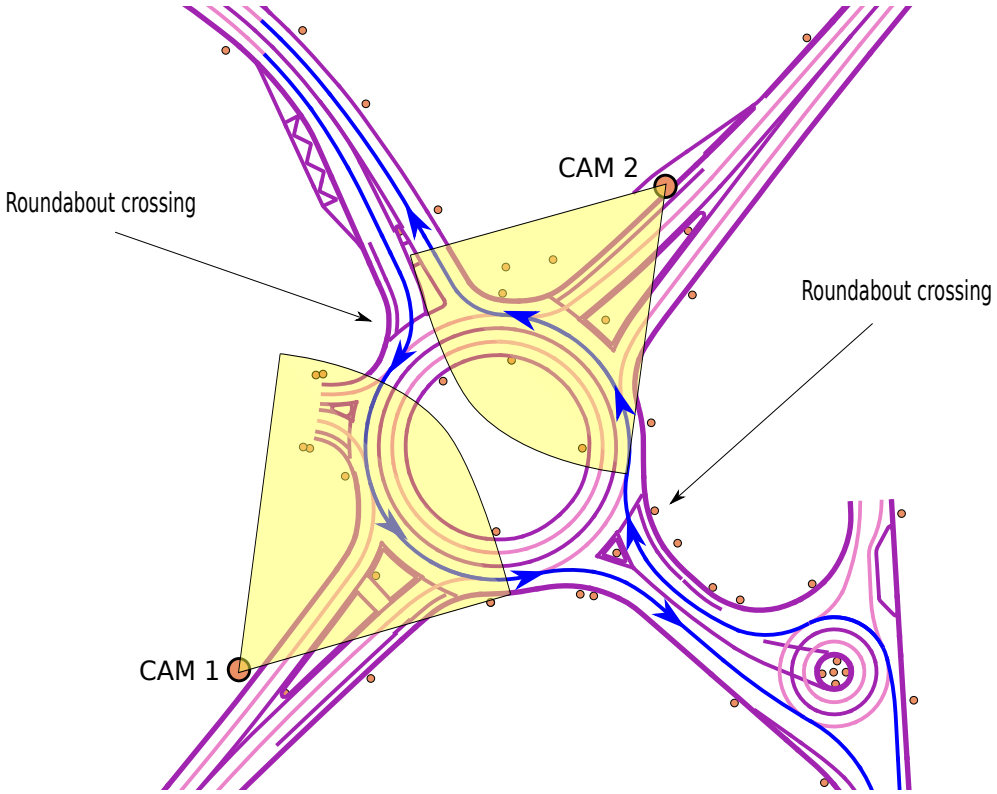


Figure 5.8: Location of the two cameras and their fields of view (yellow cones) w.r.t. the AD vehicle path (in blue) in both driving directions. City of Rambouillet, France.

Figure 5.9: An intelligent camera installed on a road pole close to the roundabout test-bed in Rambouillet. In order to obtain the best detection performance, we tested two different models of cameras that are visible on the pole.

Figure 5.10: Bounding boxes of detected vehicles from the image acquired by the intelligent infrastructure. Université Gustave Eiffel results.

algorithm YOLO (You Only Look Once) version 3 has been used [45]. This approach performs both objects detection and classification. Furthermore, a single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation, which allows this algorithm to provide detection results in a reasonable amount of time for real-time purposes.

In order to obtain more precise results in terms of classification of the detected objects, the last layer of the neural network behind YOLO has been fine-tuned using a dataset containing several driving sequences recorded in the roundabouts test sites introduced in section 2.5. Moreover, the 80 classes of objects from the COCO dataset [59] used to train YOLO have been reduced to include only the classes relevant to driving scenes.

Figure 5.10 depicts the results of the objects detection algorithm. As one can see, the different vehicles have been well detected and a bounding box has been drawn for each of them.

5.4.3 HD-Map Projection

In order to exploit the bounding boxes detected by the camera, one needs to transform them from the image frame to the HD map frame. This is needed to provide the results in the same working frame as the AD vehicle on-board perception. In particular, the infrastructure exploits the CPM (Cooperative Perception Message) standard (7) to encode the perception information and share it with the AD vehicle. This standard expects the encoding of the perceived information in a common frame between the different perception sources. In order to cope with that, the hypothesis that both the intelligent infrastructure and the AD vehicle share the same HD map-based reference frame has been done. This frame is used as a common frame to project all the perception results.

By calibrating the camera and assuming that the ground is planar, one can estimate the homography matrix that converts any pixel coordinates on the ground to its coordinates in the HD map. A first approximation of the occupancy of the detected vehicles is to consider the projection of the four corners of each bounding box. However, if one look carefully at the bounding boxes obtained in Figure 5.11,

Figure 5.11: Some examples of relative orientations of bounding boxes in the plan of the image. The bounding box in circle A covers the width and the depth of the object, while the one in circle B represents the height and a combination between width and length. Finally, the box in circle C represents the length and the height of the vehicle. This image belongs to a data sequence acquired in the city of Compiègne.

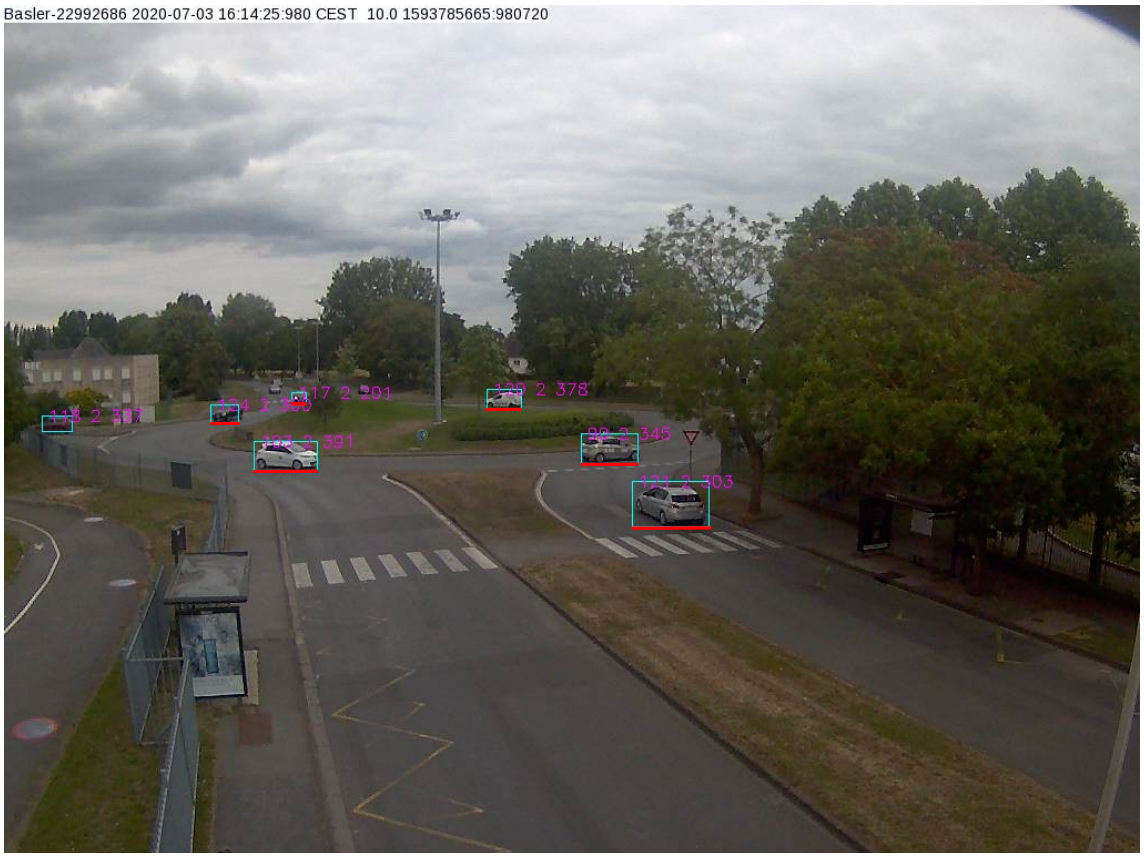


Figure 5.12: An example of bounding boxes in the image frame with their corresponding lower sides used for the bounding box projection.

one can observe that the corners of the bounding boxes do not always correspond to the planar dimension of the detected objects (length and width). In particular, this correspondence depends on the relative orientation between the detected object and the intelligent infrastructure, as it is depicted in the different cases of Figure 5.11. For this reason, a simple projection of the detected bounding boxes on the HD map plan leads to consistent but overly-estimated results. In fact, this projection tends to over-estimate the occupancy of objects, in particular their depth and does not provide information about the yaw angle of detected vehicles. To visualize this, Figure 5.13 illustrates these projections w.r.t. some detected vehicles.

To cope with that, the inferred length and width obtained from the classification information is exploited to provide a better estimate of the true occupancy of the detected object. The detection algorithm is able to classify the vehicles into different categories, *e.g.*, passenger car, truck, bus, etc., from which a standard vehicle size can be used.

Once this information has been obtained, its position is computed in the HD map frame. To do that, the red segments of Figure 5.12 are considered. These segments correspond to the lowest corners of the detected bounding boxes in the image frame (i.e. the side of the bounding box lying on the ground). These parts of the bounding boxes coincide with the projection of the vehicles on the ground plane. This information together with the information about road geometry provided by the HD map can be used to project correctly the detected MD vehicles lengths and widths into the HD map framework. The transformation is based on the information obtained from the boxes in the image frame and the prior knowledge of the HD map.

Moreover, thanks to the road geometry information, a heading angle can be inferred to orient the bounding boxes accordingly to the road where objects have been

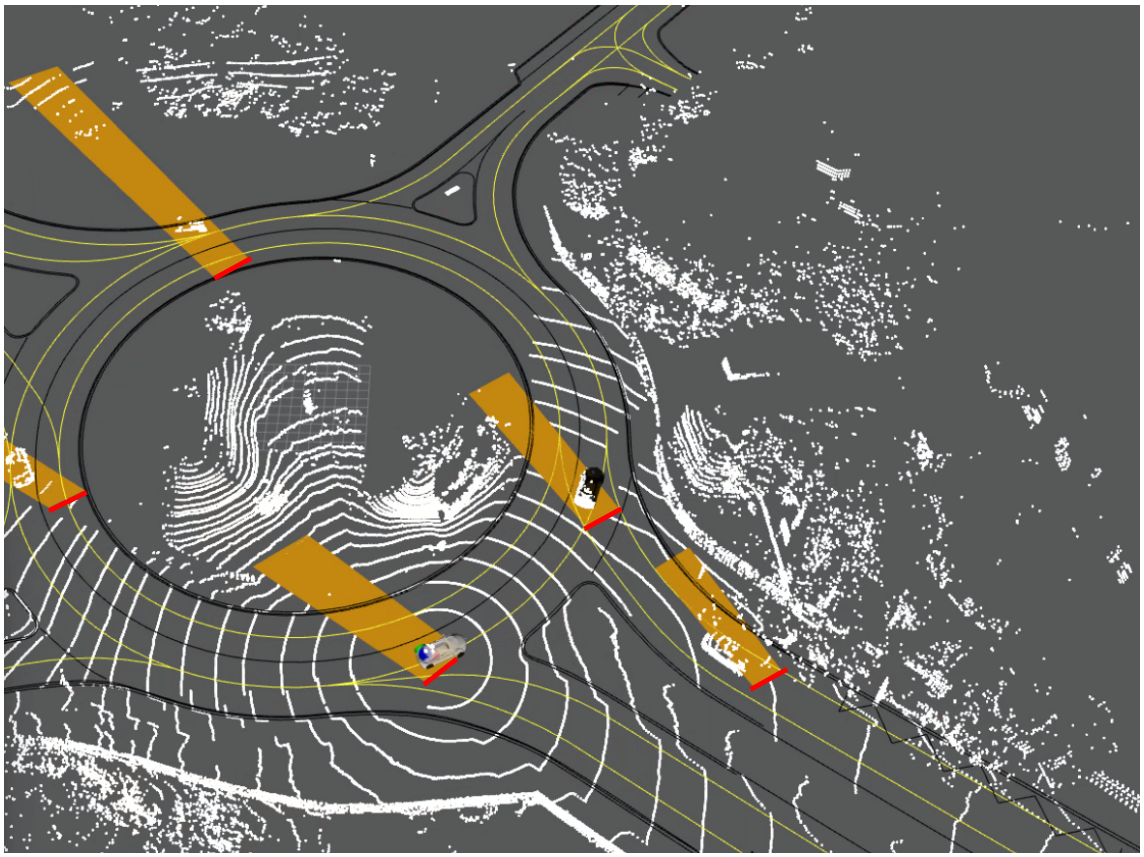


Figure 5.13: Occupancy obtained by projecting the corners of the detected bounding boxes from the image plan to the HD map plan. As one can see, overly-estimated sizes in the objects estimation occur and there is no information about the yaw angle of detected objects. The LiDAR scan of an experimental is displayed as an overlay.

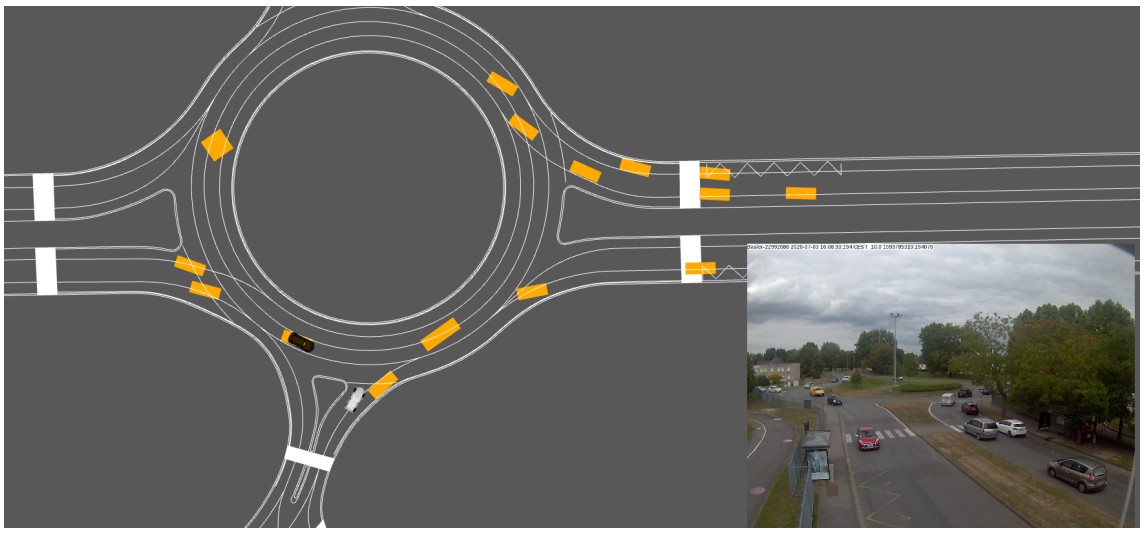


Figure 5.14: The projected bounding boxes in the HD map frame. This representation has been obtained by exploiting the classification information.

detected. This information is useful to understand the relative orientation of the detected objects w.r.t. the infrastructure and consequently to choose properly either the width or the length of the computed bounding box to represent the detected depth of the object.

To illustrate this concept, Figure 5.14 depicts the result of this new projection in the frame of the HD map. Notice that the classification results are more precise when detected objects are close to the infrastructure camera. Moreover, the localization of the objects is less precise w.r.t. the one obtained by using the LiDAR data. As one can see, the obtained bounding boxes represent the object size and they are no more dependent on the relative angle of the vehicle w.r.t. the infrastructure. Furthermore, in this infrastructure an extra layer has been added to estimate objects speeds. This layer considers the motion difference between two consecutive images, in order to compute the objects speeds. To achieve this task, a data association procedure has been implemented to retrieve detections that correspond to the same vehicle in consecutive images.

5.5 Cooperative Perception with Infrastructure

5.5.1 Problem Statement

The main objective of this section is to study how the introduction of a remote intelligent infrastructure can improve the perception of the driving environment from the AD vehicle point of view. In particular, one is interested in studying how this system can cope with the problem to perform an extended and more robust estimation of the MD vehicles position, speed, and size.

The goal is to combine the information from the intelligent infrastructure received by wireless communication with the one provided by the embedded LiDAR. In particular, we are looking for a method that allows to combine together both pieces of information in order to better estimate objects occupancy on the polylines for a safe roundabout crossing.

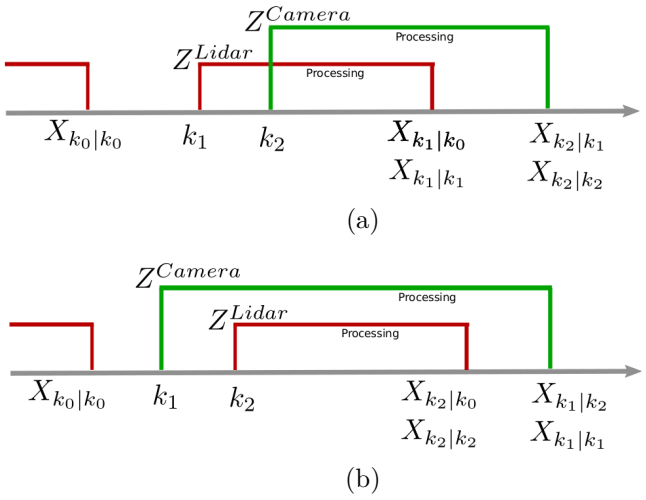


Figure 5.15: Illustration of an out-of-sequence measurements problem with an on-board LiDAR sensor (in red) and the intelligent infrastructure (in green). Figure 5.15a depicts a case where data fusion is performed in a synchronous way. In this case, the state update is performed accordingly to the chronological order of the data ($k_1 < k_2$). Figure 5.15b illustrates a situation where an out-of-sequence measurement occurs because the update is performed before considering the LiDAR sample which has an acquisition time smaller than the camera one. As a consequence, when the camera measurement arrives, it is necessary to perform a state prediction backward in time from k_2 to k_1.

5.5.2 Out-Of-Sequence Problem

A challenging problem when performing multi sensor data fusion is the synchronization of the data received from several sensors that are not working at the same rate. This problem is called "out-of-sequence data synchronization" and occurs when data sent from a sensor are received with some delay because of the computation time and/or transmission delay. Figure 5.15a illustrates the reception and processing of observations coming from the on-board LiDAR and from the infrastructure camera. In this case, we assume that both sensors provide data at the same frequency (in our case 10Hz) and that the processing of the data is completed before the next time instant.

Let us consider first the synchronous example depicted in Figure 5.15a. In this case, a previous state estimate $\hat{X}_{k_0|k_0}$, and the reception of the LiDAR data at time $k_1 > k_0$. In order to perform the data fusion between the state estimate and the received information timestamped at k_1, one first need to extrapolate the estimate $\hat{X}_{k_0|k_0}$ up to time k_1 which gives $\hat{X}_{k_1|k_0}$. The update of the state provides $\hat{X}_{k_1|k_1}$. In the meantime, the AD vehicle gets an observation from the intelligent infrastructure. Such an observation has been timestamped at $k_2 > k_1$. The information updated after the camera is computed w.r.t. the time instant $k_2 > k_1$. In order to incorporate the new measurements to the current estimate $\hat{X}_{k_1|k_1}$, the same procedure is repeated. Firstly, we extrapolate $\hat{X}_{k_1|k_1}$up to k_2, obtaining $\hat{X}_{k_2|k_1}$, then, performing the update to obtain $\hat{X}_{k_2|k_2}$. Notice that both times k_2 and k_1 correspond to times at which the observations have been timestamped by the sensors which is, in general, different from the times when the information is received by the AD vehicle.

This time difference can be caused either by the processing time or because of some latency that occur when sending information from the intelligent infrastructure to the AD vehicle. In this case, it may happen that an information that has been

acquired before is available after another one. Figure 5.15b illustrates this case. Let us consider again the case of an infrastructure data and a LiDAR acquired respectively at times k_1 and k_2, with $k_1 < k_2$. In order to be coherent with the aforementioned working hypothesis, we expect the camera data available before than the LiDAR ones. However, this time the LiDAR data, which has been acquired after the camera data, is available before the camera ones, as one can see in Figure 5.15b. To deal with this kind of scenario, one can compute again the extrapolation of $\hat{X}_{k_0|k_0}$ up to time k_2, obtaining $\hat{X}_{k_2|k_0}$ The corresponding update leads to $\hat{X}_{k_2|k_2}$.

Nonetheless, when the camera information is available to the AD vehicle, the information has been acquired at a previous time w.r.t. the current state estimation $\hat{X}_{k_2|k_2}$. In this case, with $k_1 < k_2$ we have an out-of-sequence data. In order to treat this information, several approaches are possible.

On one hand, it is possible to perform an extrapolation of the state $\hat{X}_{k_2|k_2}$ backward in time down to k_1 and compute first $\hat{X}_{k_1|k_2}$ and then the corresponding update $\hat{X}_{k_1|k_1}$. This approach allows to incorporate the out-of-sequence information in our estimation process. However, this extrapolation back in time is valid only for small delays of out-of-sequence measurements and it is not possible to use it in a general case [10].

On the other hand, it is possible to implement a buffer that keeps track of the consecutive state estimates over a fixed time slot. This buffer allows to choose the correct state estimate according to the received measurements timestamps. This allows to merge each measurement with the good state estimate from a temporal point of view. Once every measurement has been associated and fused with the correct state estimate, an overall estimation process of all the partial state estimates is carried out to provide a global estimation of the content of the buffer.

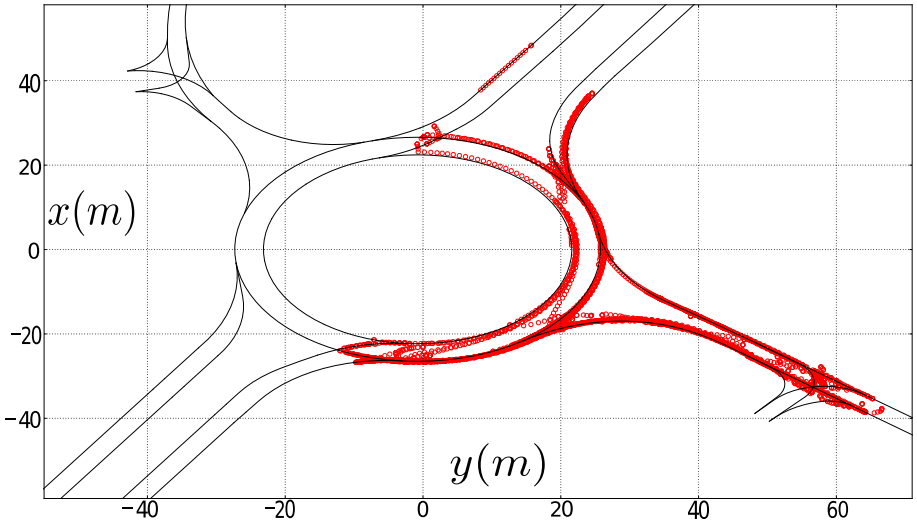
5.5.3 Data Fusion Evaluation

The asynchronous data fusion technique between the two sources of information has been evaluated experimentally. As said before, the Extended Kalman Filter technique has been used to fuse data. In order to solve data asynchronicity, The backward propagation models for back propagation of the state estimation described in section 5.5.2 Has been implemented. The whole system has been implemented in C++ with ROS environment. We report here experiments carried out with experimental vehicles.

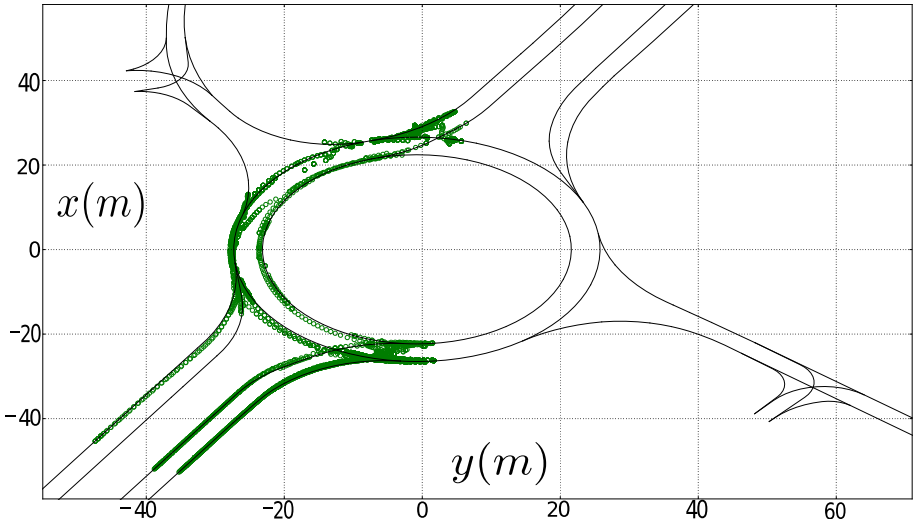
Figure 5.16 illustrates the comparison between the tracking performed first only with LiDAR data (Figure 5.16a), then with only infrastructure measurements (Figure 5.16b) and finally with both (Figure 5.16c). The data fusion has been performed at a roundabout, where the AD vehicle has been stopped in order to test its evaluation of the ongoing driving scenario before an insertion maneuver. On Figure 5.16, the red and green dots indicate the location of the the tracked road users with the on-board LiDAR sensor and with the intelligent infrastructure. The blue points correspond to a tracker that uses both modalities.

One can observe that the two sensors provide complementary information due to their different fields of view. This is highlighted by the results that are present in both Figures 5.16a and 5.16b. The zones where the different sensors track the road users are complementary and partially overlapped.

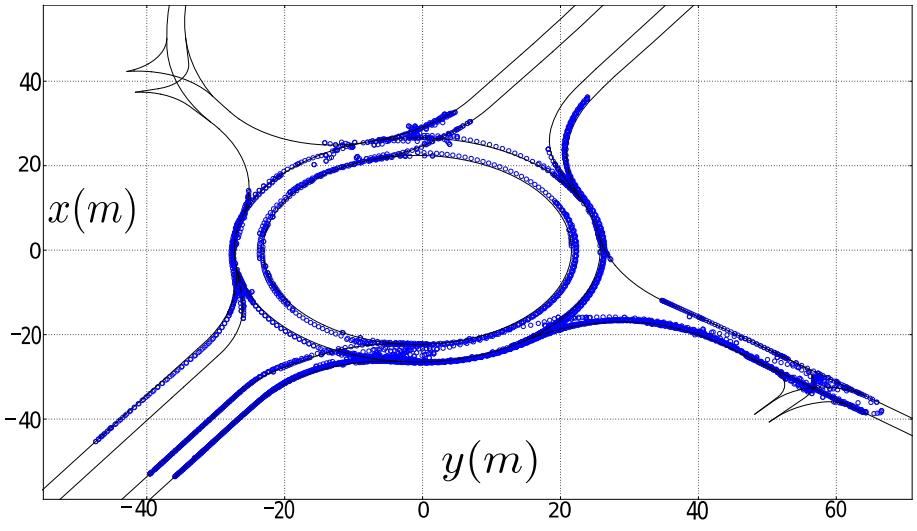
This leads us to investigate whether it is interesting for our application to maximize the field of view of the cooperative system by disposing the sensors in a configuration that enhances complementarity in their fields of view or, if it is more



(a)



(b)



(c)

Figure 5.16: Comparison of the tracking results for LiDAR (Figure 5.16a), camera (Figure 5.16b) and both (Figure 5.16c). Notice that the field of view augments in the multi-sensor case.

convenient to put them in a configuration where a consistent overlapping zone is present. The presence of the overlapping zone between the two sensors can be useful to improve the accuracy of the detected objects estimation when switching from one sensor's field of view to another. For this reason, in the following part of the chapter we will analyze in detail the different benefits and drawbacks of this configuration in terms of both enhancement of the field of view and accuracy of the state estimation.

5.5.4 Proposed Solution

Given the sensors configuration, we now investigate the best way to fuse information in the cooperative system. This new approach has been designed to exploit the main characteristics of each sensor. The LiDAR can provide a good precision in tracking objects position and speed. However, LiDAR measurements can have difficulties to estimate the length of the detected vehicles due to variation in the appearance of perceived vehicles.

The intelligent infrastructure on its side is less accurate to localize vehicles because of the camera's resolution. Nonetheless, it is able to infer objects dimensions thanks to the classification information provided by the YOLO processing of the camera images. In order to exploit the advantage of both sensors, we have therefore decided to implement the following approach:

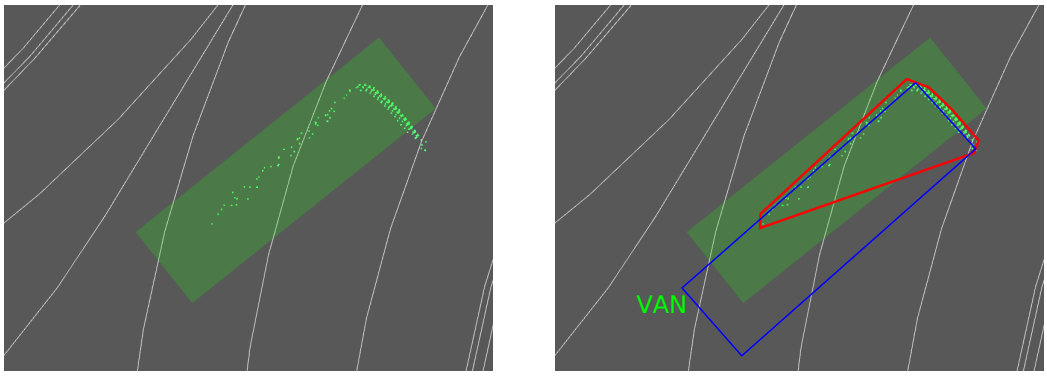
1. A multi-sensor data fusion approach which exploits both camera and LiDAR data for estimating objects state;

2. Objects occupancy is estimated by standalone sensors measurements in the zones with complementary field of view. Camera observation can be used to improve LiDAR objects occupancy estimation in the zones where overlaps occur.

Condition (1) allows to extend the LiDAR field of view by providing additional information from the cameras of the intelligent infrastructure. In this case, when no LiDAR information is available, we use the camera data to estimate the state of perceived objects. This allows to notify to the AD vehicle the presence of incoming vehicles that are not already into its on-board sensors field of view.

Condition (2) aims to introduce the camera information in the occupancy estimation process of the vehicles that are within the on-board sensor field of view. This allows to better estimate the size of perceived objects by introducing the classification information provided by the camera in the occupancy estimation process. If only camera measurements are available, they will be used to estimate the occupancy of the perceived objects.

Figure 5.17 presents our objective. It illustrates the superposition of a LiDAR point cloud and a camera detection after having solved the out-of-sequence problem. As it can be seen, the camera detection is less accurate compared to the LiDAR. However, thanks to the classification information provided by the camera, it is possible to infer the size of the object bounding polygon. The result is displayed on Figure 5.17b. In this situation, the box corresponding to the vehicle is better estimated thanks to the camera.



(a) (b)

Figure 5.17: Contribution of the intelligent infrastructure information in compensating missing LiDAR information on an object dimension. Figure 5.17a: camera detection and corresponding LiDAR measurements. Figure 5.17b: comparison of size estimation for both LiDAR (red) and camera (blue). The green text "VAN" indicates the classification information provided by the infrastructure.

5.6 Experimental Results

In this section, we present experimental results for both the LiDAR-based perception and the cooperative data fusion presented before. In particular, we focus the attention on the performance of the single LiDAR-based tracker and the infrastructure-based cooperative perception system used to estimate the size of perceived objects. The main idea is to compare the performance of both systems in terms of integrity of perceived objects occupancy estimation. This evaluation is carried out thanks to a dataset that provides the ground truth position and size for some perceived vehicles. Moreover, we also introduce a tool that helps to visualize and quantify the integrity gain for a given estimation strategy.

5.6.1 Integrity Evaluation Metrics

Let us introduce the tools that we use to carry out an integrity analysis in the rest of this chapter. The aim of the perception system is to provide to the decision level an good estimation of the speed and an occupancy estimation on the HD map of the detected objects as close as possible to the real one. In practice, we compare the occupancy estimate with the occupancy of an experimental vehicle equipped with a ground truth localization system. In other words, we compare the occupancy interval generated by the ground truth with the one obtained with the tracker.

For a given tracked occupancy interval, we are interested in estimating the percentage of overlapping surface w.r.t. the corresponding ground truth one. To do so, let us define the two quantities d_f and d_b which represent respectively the relative position of the tracked interval boundaries w.r.t. the ground truth ones. These quantities have different signs according to the relative position w.r.t. the ground truth interval. Figure 5.18 illustrates this concept.

Once d_f and d_b have been computed, we express these quantities w.r.t. the total length of the ground truth interval l. We have:

$$i_f = \frac{d_f}{l} \tag{5.26}$$

and

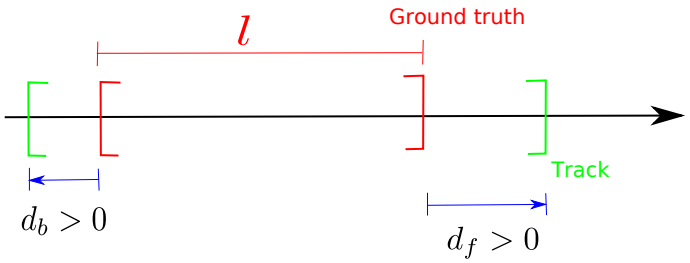


Figure 5.18: Relative position of the tracked occupancy interval boundaries (green) w.r.t. the ground truth ones (red). Please note that d_b is positive in the left direction, contrary to d_f which is positive in the right direction. In this example, both quantities d_b and d_f are positive according to the relative overlapping, while l represents the ground truth interval length.

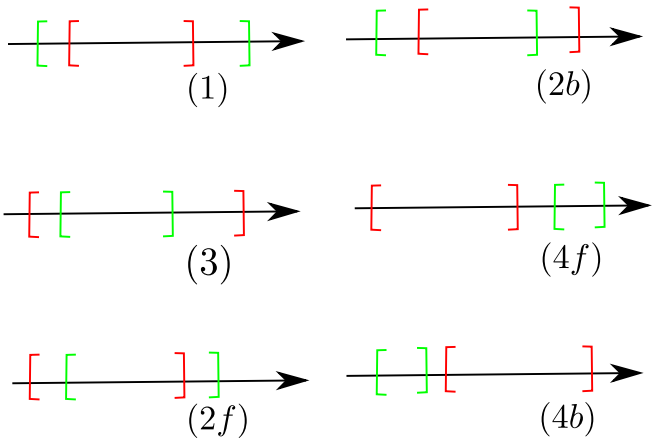


Figure 5.19: Examples of intervals compared with ground truth. The numbers correspond to the regions depicted in diagram of Figure 5.20.

$$i_b = \frac{d_b}{l} \tag{5.27}$$

According to the relative sign of i_f and i_b, one of the following situations can happen:

$$i_f = \begin{cases} 0 \leq i_f & (a) \\ -1 < i_f < 0 & (b) \\ i_f \leq -1 & (c) \end{cases} \tag{5.28}$$

$$i_b = \begin{cases} 0 \leq i_b & (d) \\ -1 < i_b < 0 & (e) \\ i_b \leq -1 & (f) \end{cases} \tag{5.29}$$

Figure 5.19 depicts examples of intervals overlapping for every case.

Equation 5.28 indicates the three possible cases that can happen regarding the forward boundary of the estimated occupancy. Case (a) implies that the boundary ahead of the tracked occupancy wraps the one of the corresponding ground truth occupancy. In other words, the integrity property is satisfied for this side of the occupancy. On the contrary, in cases (b) and (c) the estimation of the interval boundary does not contain the ground truth one. This implies that the integrity criterion is not met. In particular, case (b) means that the estimated boundary ahead is contained by the corresponding ground truth one, while case (c) corresponds to a situation where the estimated occupancy interval is disjoint w.r.t. the corresponding ground truth one.

That being said, one can repeat the same reasoning for the i_b interval considering Equation 5.29. The reasoning is similar to the previous one. In particular, case (d) represents again the only case where the integrity condition is met, while cases (e) and (f) represent respectively the underestimation of the backward occupancy boundary and intervals disjunction.

Bearing this result in mind, if one plots the quantities i_b and i_f on a 2D plane, an integrity diagram presented in Figure 5.20 is obtained. This diagram has i_f on the abscissa axis and i_b on the ordinates axis. In this diagram, the space is divided in several sub-regions according to all the possible combinations of the conditions illustrated in Equation 5.28 and 5.29. The belonging of a given point to one of those regions determines the integrity of the occupancy interval.

In the diagram of Figure 5.20, the red region represents the zone where no points in the form of (i_f, i_b) can be. This is because, if a point is in this zone, it would mean that the interval bound ahead is behind the interval bound backward, which is of course impossible. For this reason, we focus the attention on the other regions of the diagram. Zone (1) represents the area where both the front and behind occupancy interval bound are correctly estimated w.r.t. the corresponding ground truth ones. In this case, the occupancy estimation satisfies the integrity property. Moreover, the more the points are close to the i_f or i_b axes, the more the ahead and behind interval boundaries are close to the ground truth ones. The ideal case is the point $(0, 0)$. This means that the estimated occupancy overlaps perfectly the ground truth one. Conversely, as we move towards infinite, the estimates of the forward and behind intervals boundaries grow and become overly-estimated w.r.t. the corresponding ground truth.

The region that corresponds to case (2f) includes points belonging to detection where only the ahead boundary of the occupancy interval satisfies the integrity

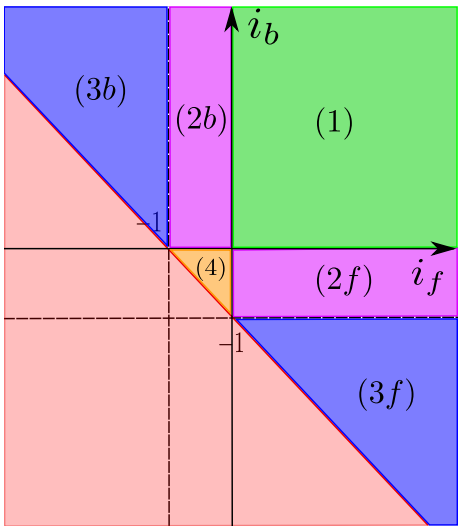


Figure 5.20: Integrity diagram. Notice that no points can belong to the red zone. Zone (1) represents a correct integrity estimation for both backward and ahead interval bounds, while zone (4) represents the under-estimation of both of them. Zones (2b) and (2f) correspond to zone where only the backward or the forward boundary satisfies the integrity property, while zones (3f) and (3b) correspond to disjoint intervals.

property. Again, as we move towards 0, the forward boundary becomes closer to the corresponding ground truth one. Regarding the backward occupancy boundary, it is always under-estimated in this region.

The dual situation of case (2f) is represented by case (2b). In facts, this region includes points that satisfies the integrity property for the behind interval boundary but not for the boundary ahead.

Regions (3b) and (3f) represent zones in which the intervals are disjointed. In case (3b) the estimated occupancy interval is behind the corresponding ground truth one, without any intersection and, in case (3f) the estimated occupancy is ahead the ground truth one. For none of those cases we consider the integrity property satisfied.

Finally, the region represented by case (4) contains points that have both the forward and backward interval boundary under-estimated w.r.t the corresponding ground truth one. Again, none of the points belonging to this region satisfy the integrity property.

Notice that, in comparison with cases (3b) and (3f) points that belong to region (4) have the advantage that, even if they do not satisfy the integrity property, the resulting estimated interval provides a clue about the position of the estimated object. In fact, the estimated occupancy is contained entirely inside the ground truth one, which is not the case for points in (3b) and (3f), where, if the object has been wrongly detected, the corresponding occupancy can be very far from the object ground truth.

5.6.2 Experimental set-up

In order to validate the algorithms presented before, we exploit data collected with experimental vehicles platform in the city of Compiègne. For this experiment, we have used a dataset that contains recorded accurate positions of the AD vehicle (white car) and another vehicle (grey car) used as an obstacle. For both cars, the accurate position were obtained with a Novatel Span CPT GNSS receiver, with

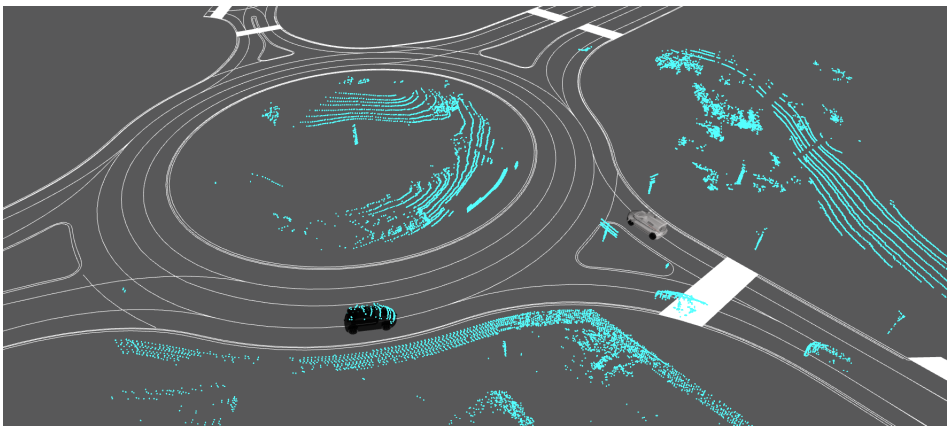


Figure 5.21: The AD vehicle used for the data acquisition (grey vehicle) and another vehicle of the platform used for ground truth localization (in black).

Post Processed Kinematics (PPK) corrections, which gives a centimetric-level of accuracy. Moreover, the white vehicle was equipped also with a Velodyne VLP32 LiDAR sensor used to provide the observations for the LiDAR-based objects tracking system.

In the experiment, we used the white car to simulate the AD vehicle navigation through a roundabout, stopping also the car at the roundabout entrance to acquire some data about the incoming traffic flow. At the same time, the grey car was navigating into the roundabout among other vehicles, providing extra obstacles for the white car. The main advantage of this approach is that when the white car detects and tracks the gray one, it is possible to compare the resulting estimated state of the track (e.g. position and velocity) w.r.t. the ones obtained with the ground truth localization system. To do so, we select, for every time instant and among all the detected objects, the one that corresponds to the track of the gray vehicle and we compare its estimated state and occupancy with the ground truth ones. To visualize this, Figure 5.21 illustrates a scenario of the dataset with the two cars and the corresponding LiDAR point cloud.

Regarding the infrastructure system, we have installed a fixed camera on a bridge in the southern direction of the roundabout, in order to observe the traffic on that side of the roundabout. The camera was synchronized with the clocks of the on-board systems of the two cars (through GNSS) and it recorded images about the vehicles in the roundabout. After the recording session, the camera data were sent to the Université Gustave Eiffel and treated offline to detect vehicles in the recorded images and to provide the vehicles position and occupancy estimation in a CPM like format.

Figures 5.22 and 5.23 depict the navigation scenarios for this validation tests with the corresponding trajectories of the AD vehicle.

5.6.3 Lidar-based Tracking and Objects Occupancy Estimation

In this section, we detail the experiments that we carried out on real data to test the performance of the LiDAR-based tracking system. To do so, we have implemented in C++ the tracking algorithm and we have run the tracker on the testing scenarios described previously.

Figure 5.24 illustrates the results of the tracking system in some sequences of the dataset. In particular, the three scenarios presented in Figure 5.24a, Figure 5.24b and Figure 5.24c represent respectively three subsequent time instants and show the

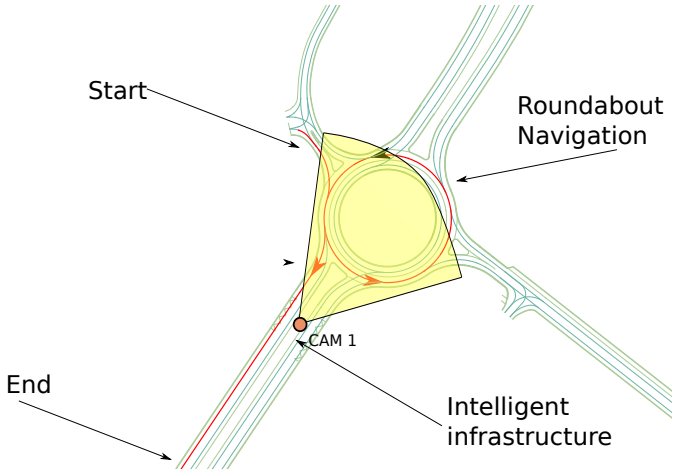


Figure 5.22: A scenario recorded in the dataset. The AD vehicle trajectory is in red and the yellow cone represents the infrastructure field of view.

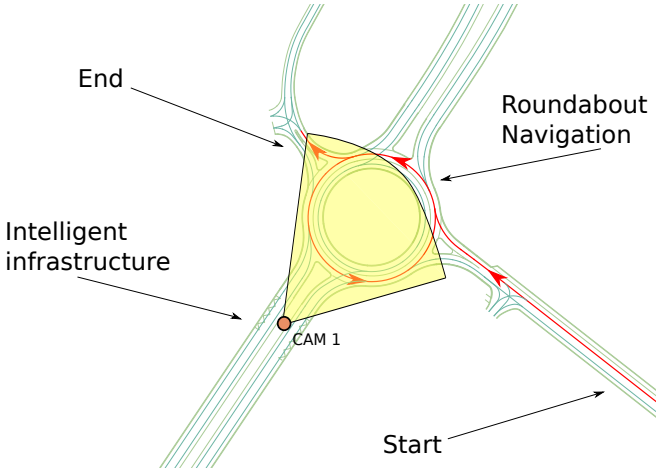
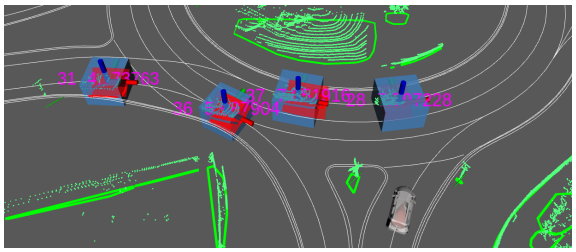
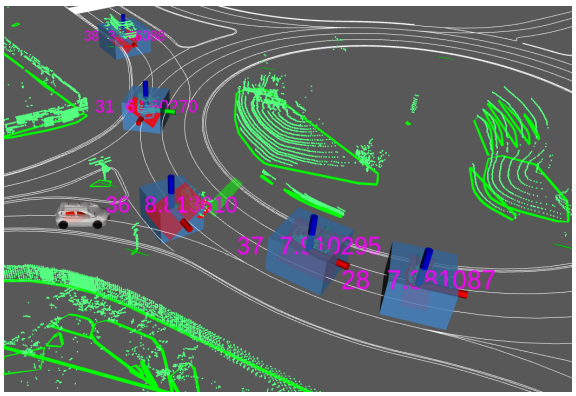


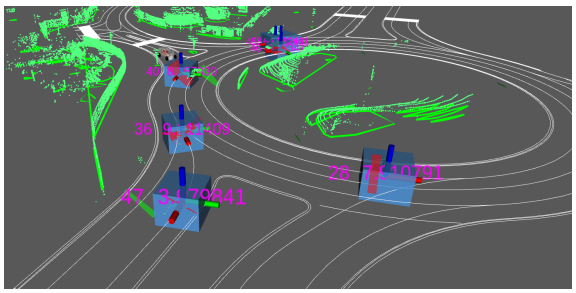
Figure 5.23: Another scenario recorded in the dataset. The AD vehicle trajectory is in red and the yellow cone represents the infrastructure field of view.



(a)



(b)



(c)

Figure 5.24: Tracked objects. The blue green and red local frame represents the pose of every tracked object, while the blue boxes represent their occupancy estimation. The LiDAR detections are in green and the corresponding curvilinear occupancy is represented by a red segment. Only the objects belonging to the drivable surface are tracked.

evolution of the tracks estimates over this time horizon.

In this scenario, the white car represents the AD vehicle, while the magenta text over each object represents the object label and the estimated object speed. Notice that, in many cases, objects keep the same label throughout the whole time horizon, while in Figure 5.24c the object labeled as "28" keeps the previously estimated size even if the corresponding observed occupancy is smaller.

5.6.3.1 Analysis of the Lidar-Only Tracking

The main objective of this section is to analyze and discuss the performance of the tracking system. To do so, we exploit the ground truth data described in section 5.6.2 to carry out a more quantitative analysis of the algorithm performance.

Figures 5.25 and 5.26 illustrate the estimation error relative to the position estimation in the x and y directions for the whole duration of the data sequence. These errors are plotted with a $\pm 2\sigma$ envelope for the error bounds, which is represented

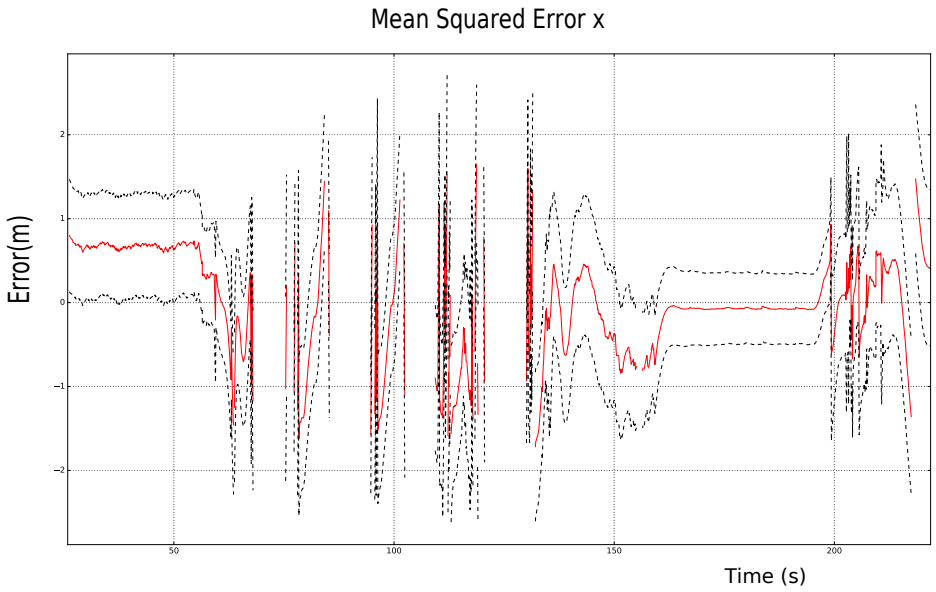


Figure 5.25: Estimation error of the x-coordinate as a function of time and the corresponding $\pm 2\sigma$ uncertainty bound.

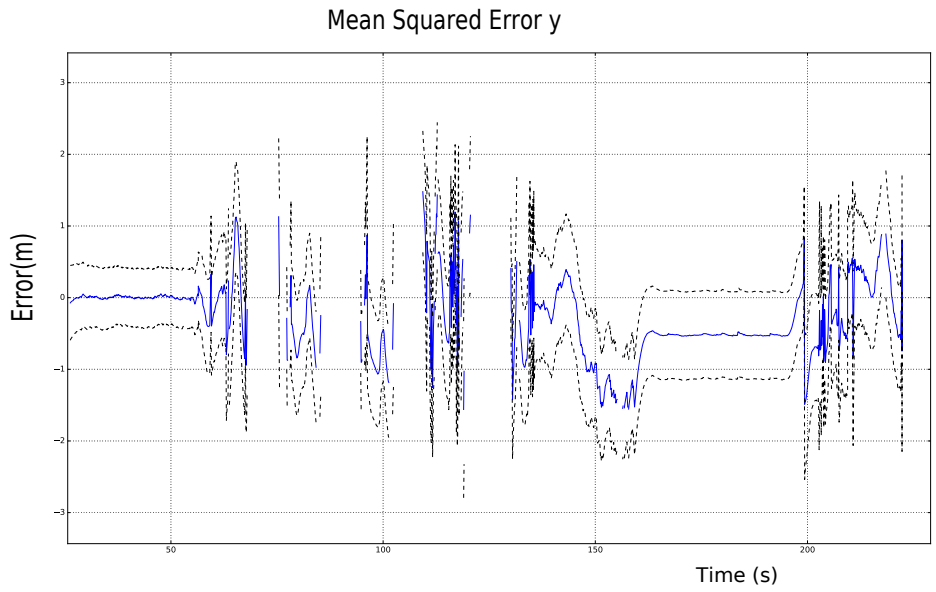


Figure 5.26: Estimation error of the y-coordinate as a function of time and the corresponding $\pm 2\sigma$ uncertainty bound.

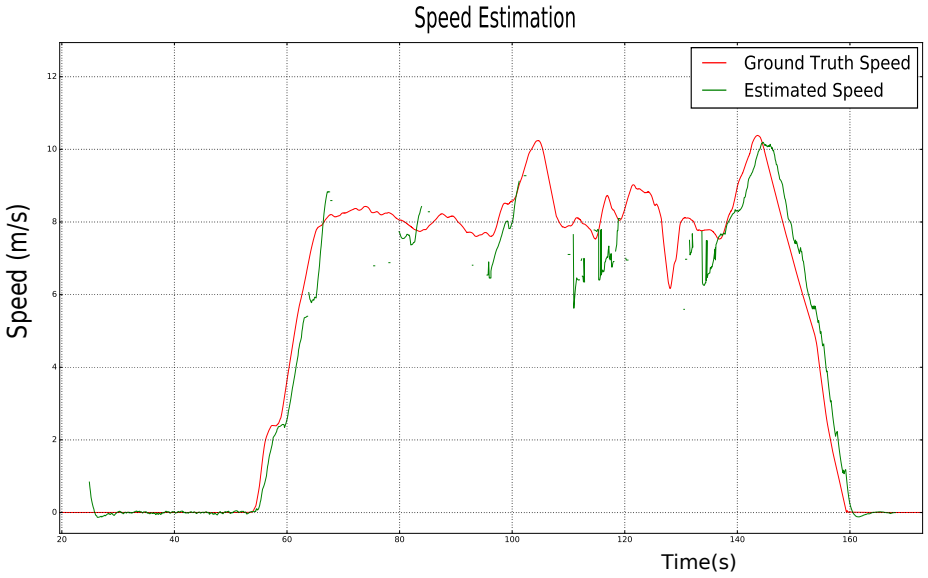


Figure 5.27: Tracked speed estimation (green) and the corresponding ground-truth speed (red). Notice that the speed is estimated only when the vehicle is perceived by the system.

by the dashed black bounds and it is computed comparing the obtained estimation of the gray vehicle position with the corresponding position obtained by the ground truth data. For both figures, only the points where the ground truth vehicle is perceived by the LiDAR-based perception system are represented in the plots. Notice that in both figures we observe that the estimation of the position is more stable when the vehicle is stopped. This can be seen in the samples of the intervals around $[0, 50]$ and $[160, 200]$. This is due to the fact that the shape of the perceived object does not change significantly between each time step. On the contrary, we can observe that when the vehicle is performing rounds of the roundabout in the interval around $[60, 140]$, the estimation becomes more imprecise due to the different shape of the perceived object bounding polygons. This shape difference comes from the fact that the object is perceived from different points of view by the AD vehicle that is waiting at the roundabout entrance.

Regarding the resulting tracked speed, Figure 5.27 provides the speed estimation w.r.t. the speed profile recorded by the ground truth. Again, the speed estimation is visible only when the vehicle is detected by the tracking system. An unavoidable delay in speed estimation can be observed. On the other hand, the estimation is not too noisy. There are also areas where the object is not followed because of masking in the roundabout.

Finally, to visualize the resulting tracked paths, Figure 5.28 illustrates the estimated tracks of the ground truth detection and the corresponding ground truth trajectory, which is recorded in the dataset.

The color of the point corresponds to the labels given to the track. If the color changes, this means that the tracker has lost the track and it needs to re-initialize another one which is compared to the ground truth car position and speed. Moreover, we need to underline that the estimated track position, i.e. the red solid lines of Figure 5.28 are estimated following the polylines of the HD map. For this reason, sometimes the estimate can be different from the ground truth one (the black dashed lines). This happens because of the natural way of driving a vehicle, which does not always follow straightforwardly the road lanes.

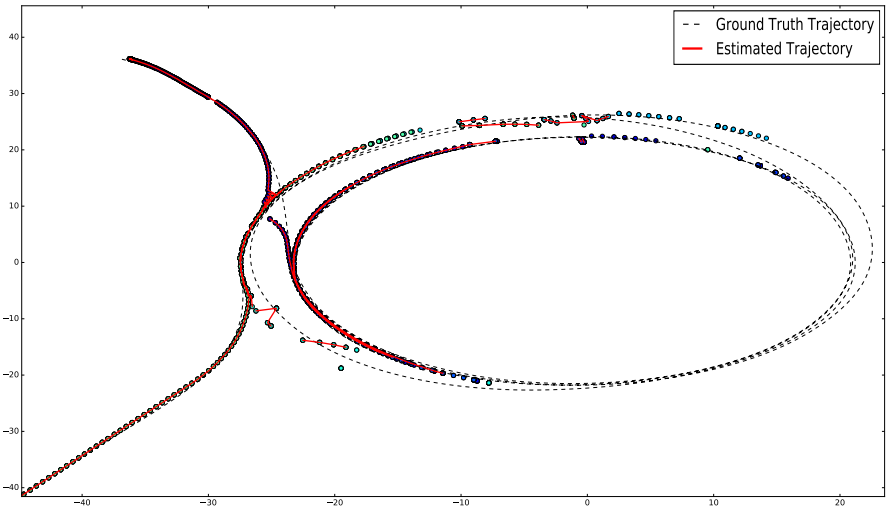


Figure 5.28: The tracked position of the gray car (red) and its corresponding ground truth path (dashed black lines) A change of color of the points in the plot represent the creation of a new object to track the ground truth car.

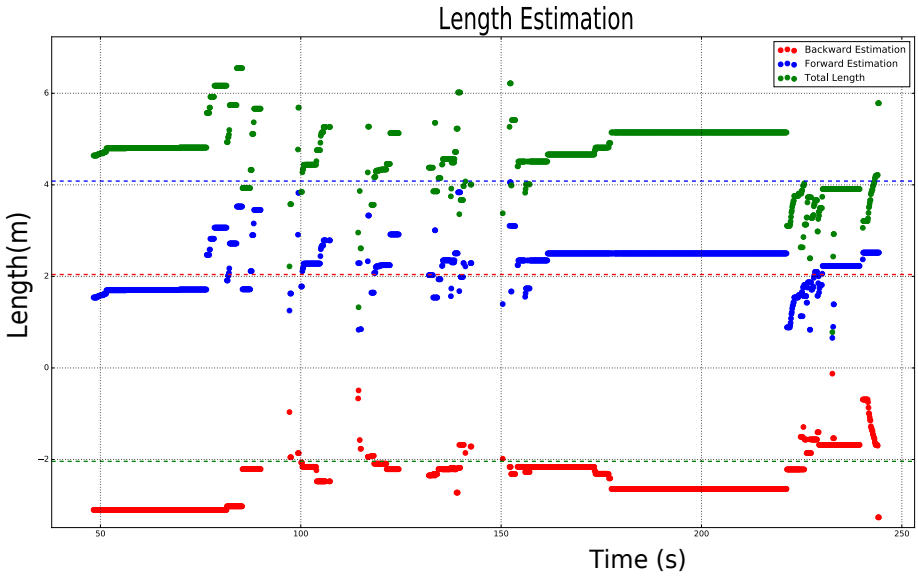


Figure 5.29: Estimation of the vehicle length (green) and the corresponding behind (red) and ahead (blue) boundaries compared to the ground truth length of the vehicle which is 4.082 m.

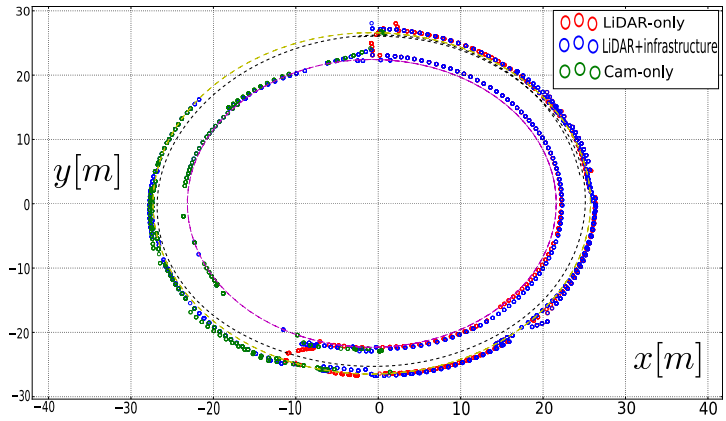


Figure 5.30: Comparison of the performance of the three tracking systems (red, blue and green dots) in terms of field of view extension w.r.t. the ground truth data (dashed black line).

Finally, Figure 5.29 gives the estimate of the vehicle total length and its forward and backward occupancy interval boundaries. As one can see, the blue total length is often above the ground truth size, which is represented by the blue dashed line. This implies that the estimated size of the object is often overly-estimated w.r.t. the true one.

All these results show overall that the LiDAR tracker works well enough to estimate the position of the tracks as well as their speed.

5.6.3.2 Analysis of the Cooperative Tracking

The main objective of this section is to analyze and compare the performance of the cooperative tracking system regarding the state estimation accuracy. We evaluate the performance of the tracking system when it uses the on-board and infrastructure data separately and then combined together.

To illustrate the accuracy of the tracking system, Figure 5.30 depicts the tracking result for only a target vehicle that was accurately localized with a post-processed SPAN CPT IMU. It shows the tracking results using only the LiDAR data, the camera data and both compared with the ground truth of the vehicle. From this figure, one can notice that even if the target car is free to move in the 2D space of the map, the tracking result is more constrained to be close to the HD map polylines. This is because of the map-aided tracking method. However, one can notice some cases where the tracking result deviates from the polylines. This can happen for example in the case of a lane change maneuver.

Furthermore, we have computed the root mean squared error for the three cases described before. To do so, we have projected both the estimated state and the ground truth position of the vehicle on the roundabout polylines and compared the estimation error in terms of along track error in curvilinear coordinates. This is useful because, many autonomous vehicles navigation and motion planning strategies rely on curvilinear coordinates [68]. For this reason, it is interesting to investigate the accuracy in estimating objects in a curvilinear framework. Table 5.1 illustrates the computation of the RMSE for the three cases. When using the whole sequence, the LiDAR-only tracking performance is better than the camera-only one. This is mainly due to the different accuracy of the sensors. The combined approach behaves as a compromise between the two. When focusing the analysis on the parts of the sequence where both the AD vehicle and infrastructure can track the target vehicle, the combination of both sensors performs better than the camera-only and

Table 5.1: Root mean square error and the duration of the time sequence (in seconds) for the tracking methods obtained using only the LiDAR, the camera and both sensors.

	Class Changing			
Sensors	Whole (m)	Time (s)	Overlapping (m)	Time (s)
LiDAR	0.70	31.5	0.99	4.3
Camera	1.88	17.1	0.78	4.3
Both	1.31	42.5	0.49	4.3

the LiDAR-only ones. In this particular situation, which corresponds to the zones where we have a transition from the camera-only setup to the LiDAR-only one, the target vehicle is almost out of the field of view of the sensor and perceived from a frontal orientation w.r.t. the LiDAR, making the estimation of its size less accurate.

5.6.3.3 Occupancy Integrity Analysis

After having presented the performance of the tracking system for both estimating the state of the detected vehicles, we focus our attention on the integrity estimation of the vehicles occupancy. Again, we exploit the ground truth data collected in the dataset as we previously explained in section 5.6.2 to analyze in a quantitative fashion the results.

Let recall that the final occupancy estimation is obtained by the tracked occupancy estimation and the injection of the estimated state uncertainty, following the method discussed in section 5.3.8. Figure 5.31 illustrates the obtained estimated occupancy boundaries compared with the corresponding ground truth ones. Notice that for a given estimation of the ahead occupancy boundary (blue), we consider that this estimation satisfies the integrity criterion if it contains the true occupancy interval boundary. In other words, if a blue point lies above the blue line, this means that it satisfies the integrity criterion.

The same reasoning can be repeated for red points also, which correspond to the backward interval boundary. However, this time the condition needs to be inverted due to the negative coordinate of the local frame centered in the ground truth vehicle barycenter. This means that, if a red point lies underneath the red solid line, it satisfies the integrity property.

To obtain an occupancy estimation that satisfies the integrity property, we want that both the ahead and behind estimates satisfy the integrity property. As we introduced in section 5.6.1, we propose to use the integrity diagram to visualize when this condition is fully satisfied by the estimates. Figure 5.32 depicts the integrity diagram for all the estimates corresponding to the ground truth data. As one can see, this result shows that almost all the points are distributed between the regions (1) and (2b). This implies that in the majority of the cases the system provides an estimation of the backward occupancy interval bound that satisfies the integrity property. In many other cases, i.e. cases where points belong to region (1), the whole estimation of the occupancy interval satisfies the integrity property.

5.6.4 Cooperative Perception: LiDAR and Infrastructure

After having analyzed the performance of the LiDAR-based tracking system, we move our attention to the performance of the cooperative data fusion system. The

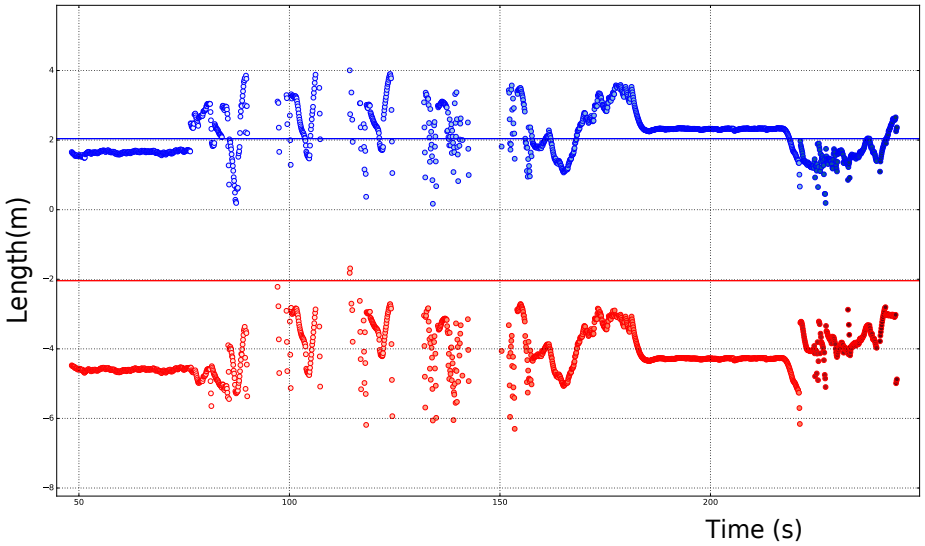


Figure 5.31: The resulting estimation of the gray car occupancy ahead (blue) and backward (red) w.r.t. the corresponding ground truth (solid lines).

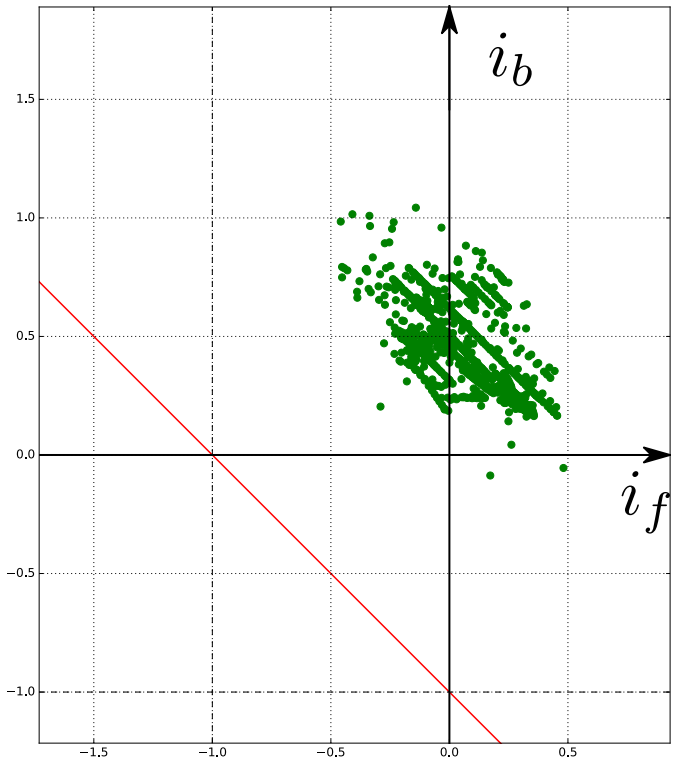


Figure 5.32: Integrity diagram for the ground truth tracks according to the region described in section 5.6.1 (LiDAR only). Notice that for both the axes the unit of measurement is dimensionless, according to the definition of i_b and i_f. The percentage of points in the zone (1) is the 55%, while the points in zone (2b) are the 45%. Finally, only the 0.001% of the points is in the (2f) zone.

main objective of this step is to evaluate the cooperative data fusion with data broadcast from the remote intelligent infrastructure, as we previously explained in section 5.5. The main interest of this step is to check whether the performance of this new cooperative system outperforms the previously obtained results. To carry out this comparison, we focus on two principal aspects: the enhancement of the AD vehicle field of view, which will be discussed in the next section and the improvement in the occupancy estimation of perceived objects, which will be analyzed in section 5.6.4.2.

5.6.4.1 Enhanced Perception Field of View

The first characteristic that we analyze here is the enhancement of the AD vehicle field of view obtained thanks to the collaborative perception system. In this case, we use again the dataset presented in section 5.6.2. This time, we exploit not only the data about the AD vehicle and the ground truth car, but also the vehicles detections obtained from the intelligent infrastructure.

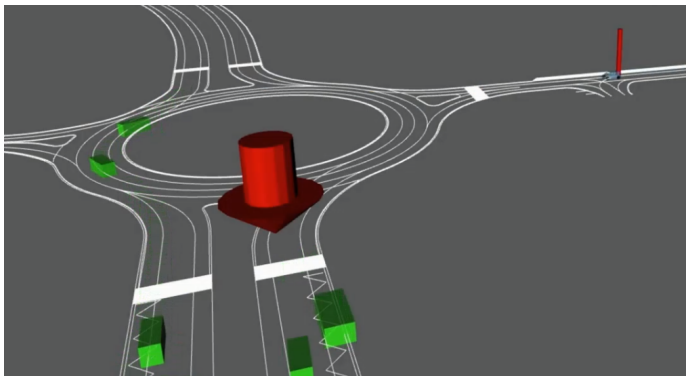
In this experiment, we use the recorded scenario to determine the impact of the intelligent infrastructure perception on the AD vehicle decision to perform an insertion maneuver. Figure 5.33 illustrates the scenario. In this situation, the AD vehicle comes from the right road and wants to enter into the roundabout. Figure 5.33a illustrates a situation where the AD vehicle can take the decision of stopping before entering into the roundabout sufficiently far ahead thanks to the perception of the intelligent infrastructure. The red arrow in Figure 5.33a indicates the detected objects broadcast to the AD vehicle. Notice that no bounding boxes belonging to the AD vehicle perception are present because the vehicle is still too far from the roundabout.

On the other hand, Figure 5.33b depicts the instant where the AD vehicle decides to perform a stop maneuver considering only its on-board perception system. As one can see, the red arrow in Figure 5.33b points towards the first object detection obtained from the on-board perception system (blue box) that causes the AD vehicle to stop at the roundabout entrance.

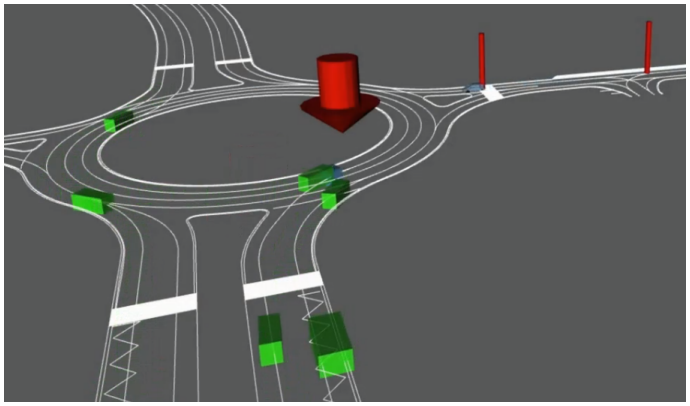
Finally, the two red markers present in Figure 5.33b illustrate the different instants where the decision of stopping before entering has been taken either considering the cooperative perception system (the farthest marker on the right) or considering only the on-board perception results (the closest marker to the roundabout).

Furthermore, we are also interested in providing quantitative results to quantify the performance gain provided by the infrastructure. Let us consider again the experiment presented in section 5.6.3.2. In this case, we consider again the tracking results depicted in Figure 5.30. To quantify the gain of the cooperative system w.r.t. the two single-sensor ones, we compute the percentage of time where the target vehicle has been tracked by each of them. This percentage is computed as the ratio between the number of samples that correspond to the target vehicle and the total number of samples of the ground truth in the time interval represented in Figure 5.30. For the single-sensor systems, the LiDAR-only system tracks the target vehicle for 68% of the time, while the camera-only system tracks it for 37%. Regarding the multi-sensor tracker, it tracks the target vehicle for 92% of the total time, outperforming the two others. Notice that the sum of the single sensor system percentages does not correspond to the multi-sensor system percentage because of the presence of overlapping zones in sensors fields of view.

In conclusion, we can observe that a remote infrastructure as the one that we



(a)



(b)

Figure 5.33: Infrastructure impact on the AD vehicle insertion maneuver. The infrastructure perception is represented by green boxes, while the AD vehicle is represented by a blue box. The small red markers indicate the instants where the AD vehicle took the decision to stop at the roundabout with only on-board perception (Figure 5.33b) and with cooperative perception (Figure 5.33a).

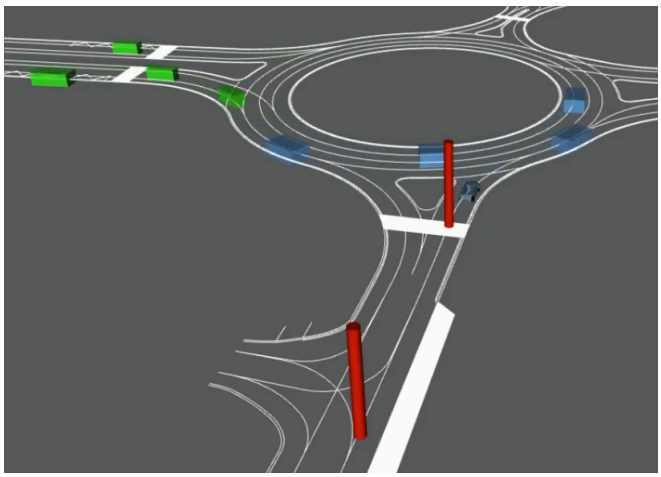


Figure 5.34: The perceived objects bounding boxes generated from the intelligent infrastructure (green boxes) and the ones generated by our on-board perception system (blue) for a roundabout insertion scenario.

used in this study can help efficiently to enhance the AD vehicle field of view. To better observe this aspect, Figure 5.34 illustrates the comparison between the boxes generated by the remote infrastructure system and the ones generated by the AD vehicle perception one. This enlargement of the vehicle field of view allows an AD vehicle to have a wider horizon to plan its driving maneuver, as we have seen in the case of the roundabout insertion maneuver.

5.6.4.2 Added Value of the Infrastructure Information with Field of View Overlap

The second aspect that we want to analyze is the improvement that the remote infrastructure can offer to better estimate the occupancy of the detected objects. By considering the proposed data fusion strategy presented in Section 5.5.4, we use again the dataset described in section 5.6.2 to perform a quantitative analysis on the performance of the system in terms of integrity estimation of the vehicles occupancy. In other words, we perform again what we previously did with the LiDAR-based on-board perception system in section 5.6.3.3, but this time considering the cooperative data fusion between LiDAR and infrastructure data.

To visualize the performance of this approach, we draw the same plots that we have shown in section 5.6.3.3 for the cooperative system. In order to properly compare the performance of this new system with the previous one, we consider only the zone where there is an overlapping field of view of both the LiDAR and the camera to perform cooperative data fusion. This means that we associate camera detection to the tracked objects obtained by the LiDAR tracker and, if an association occurs, we integrate the size information provided by the intelligent infrastructure to the estimation process.

Figure 5.35 gives the final backward and ahead occupancy interval bounds estimation w.r.t. the ground truth ones. As we can see, the introduction of the intelligent infrastructure information does not provide a remarkable improvement in the occupancy estimation compared with the one obtained by the LiDAR only system shown on Figure 5.31. Moreover, one can also observe that the integrity diagram of Figure 5.36 provides almost the same results of the previous one. However, if we have a closer look at the i_b axis of the new diagram, one can see that a few more points are lying closer to it. This is because of the added information of the intelligent

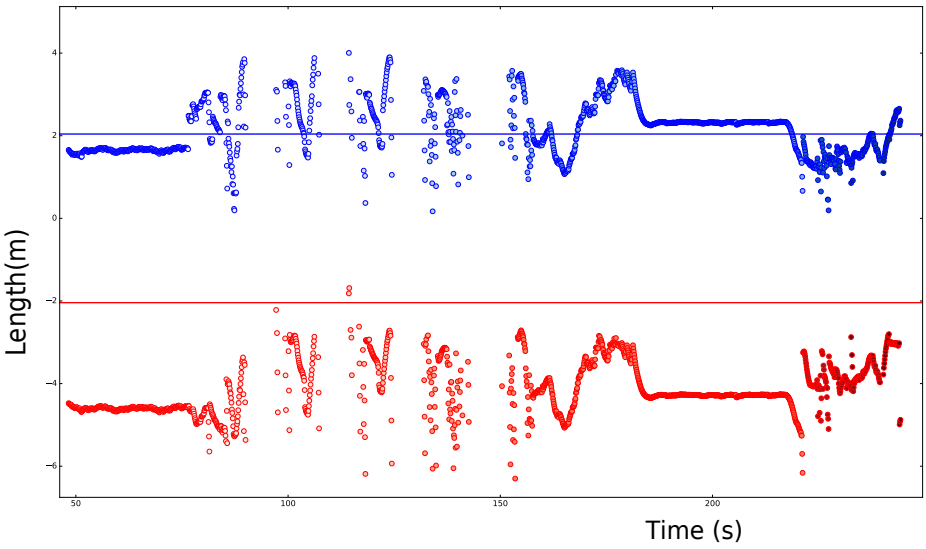


Figure 5.35: The resulting estimation of the gray car occupancy ahead (blue) and backward (red) w.r.t. the corresponding ground truth (solid lines) thanks to the cooperative perception system.

infrastructure. Nonetheless, this percentage of points is negligible with respect to the overall result. This can be due to the fact that our tracking policy for estimating the objects occupancy tends to be pessimistic. In fact, it allows the object occupancy only to grow. Moreover, the addition of the uncertainty obtained from the state estimation to the final occupancy estimation leads to a more pessimistic occupancy too and so the information provided by the infrastructure as very little impact. To better visualize that, Figure 5.37 illustrates a comparison between the results depicted in Figure 5.36 and the ones shown in Figure 5.32. Notice that in the cases where there is the influence of the infrastructure, the estimation becomes always more consistent. Moreover, we can observe that, in some particular cases, we switch from a situation where the estimation does not satisfy the integrity criterion to a situation where it is satisfied.

5.6.5 Discussion on the Added Value of the Infrastructure

In the previous sections, we have presented some approaches to propagate the localization uncertainty directly into the tracks occupancy at lane-level accordingly to the integrity concept. This method allows to model correctly the perceived objects occupancy and it leads to an efficient representation of the road users. Nonetheless, to properly evaluate the integrity of the obtained results, some considerations need to be pointed out.

Firstly, the proposed algorithm provides a good result that performs high integrity perception in terms of localization of the road users along the lanes. This represents a step towards a robust perception of the driving scene when HD maps are used for decision. However, for our specific use-case, we are interested also in guaranteeing integrity of the perceived objects occupancy on the road surface. For this reason, we have decided to focus the attention on checking whether the objects bounding polygons provide consistent curvilinear occupancy w.r.t. the real objects. To do so, we have used experimental recorded shared perception datasets. Such datasets contain the ground truth localization of some experimental vehicles. In our specific

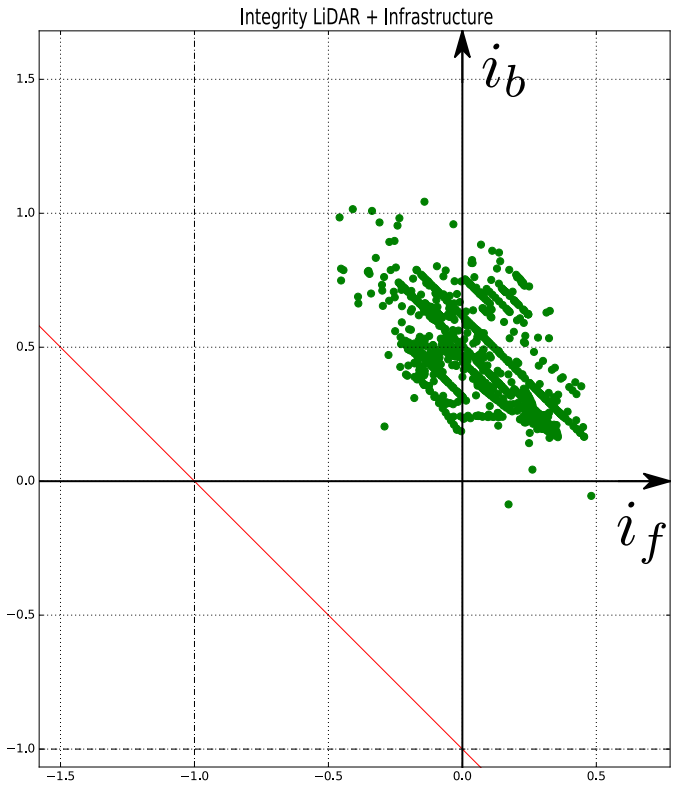


Figure 5.36: Integrity diagram for the ground truth tracks computed with the help of the cooperative perception system.

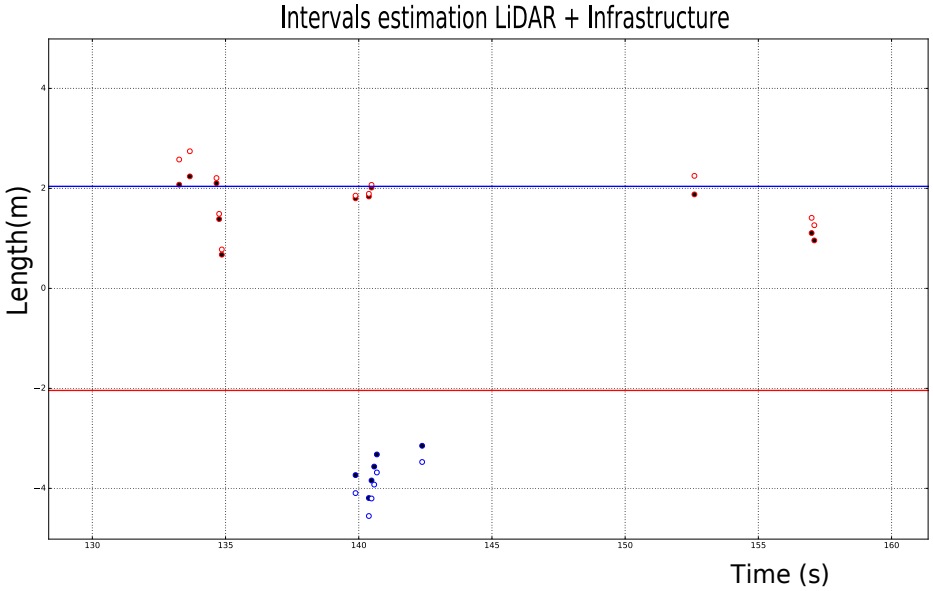


Figure 5.37: Comparison of the resulting estimation of the gray car occupancy ahead (blue) and backward (red) obtained with the LiDAR (full dots) and with the cooperative system (empty dots) in the case where the infrastructure improves the estimated occupancy. The solid lines represent the corresponding ground truth.

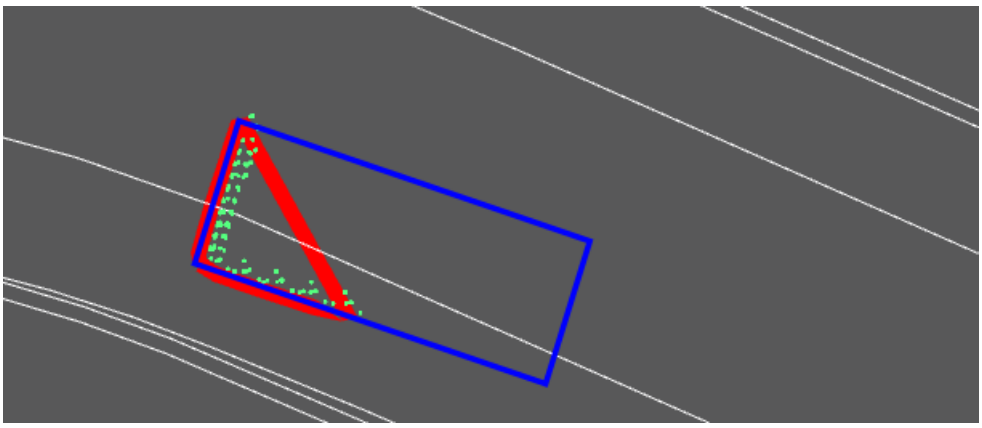


Figure 5.38: An example of a detected object (red polygon) and its corresponding ground truth occupancy (blue rectangle).

case, we used only the on-board perception of one experimental vehicle and, regarding the others, we used them as road obstacles. The main advantage is that the position of the experimental vehicles is known precisely thanks to a precise GNSS positioning system and it can be used as a ground truth to evaluate the approach. This allows to compare if the AD vehicle estimated occupancy is coherent w.r.t. the real object size. However, for the LiDAR-based perception system, one can observe that it is not always the case. This is mainly due to the fact that, in general, the LiDAR sensor detections do not respect the integrity property. This means that, for a given object, LiDAR sensors are not always able to provide a detection of the whole object. This is mostly dependent on the relative positioning of the detected object. To illustrate this concept, Figure 5.38 depicts a case where an incomplete LiDAR detection produces a partial bounding polygon. This bounding polygon does not cover all the corresponding occupancy of the detected object.

However, considering the results presented in the previous part of this chapter, we can conclude that in our particular case, the infrastructure information does not provide a significant improvement in the process of estimation of the perceived objects occupancy.

However, we observe that a cooperative system with a remote intelligent infrastructure is helpful to enhance and extend the field of view of an on-board perception system. This extension is useful to provide the AD vehicle an extended perception of the ongoing driving situation. This additional information can be used to implement a more safe and optimized navigation strategy, in particular for safety critical driving maneuvers.

5.7 Conclusion

In this chapter, we have presented a cooperative map-aided tracking system that exploit both LiDAR and camera data to estimate the detected road users state and their occupancy along the HD map polylines. The occupancy estimation is carried out by considering both the observed occupancy and the estimated uncertainty about the detected track estimated state. Moreover, the performance of this approach has been evaluated in terms of integrity w.r.t. the real occupancy of a detected object thanks to the ground truth data recorded in a dataset. More detailed results have been discussed in [65].

In the first experimental part, the performance of this system has been studied in terms of augmentation of the AD vehicle field of view, highlighting the gain of

the cooperative perception system over the standalone AD on-board perception. Clearly, a cooperative system improves of the ability to track vehicles in almost all the parts of the roundabout ring that has been considered in this work.

Then, we have presented an evaluation of the tracking performance thanks to the use of a target vehicle accurately localized which allows computing errors on real data. As observed, the three trackers have different accuracy, depending on the sensors capabilities. This means that, in this particular configuration and with the chosen sensors, enhancing the field of view of the AD vehicle by adding an extra source of information implies an accuracy loss in the overall estimation of the position of the tracked objects by the standalone AD vehicle perception. Nonetheless, we have shown that this multi-sensor data fusion leads to benefits in the zones where the field of view of the two sources of perception overlap, providing a more accurate estimation of the state of the perceived objects.

In the second part of the chapter, we have considered the map-aided tracking algorithm but with the assistance of a cooperative remote infrastructure to improve the performance of the perception. We have leveraged the classification information obtained from the remote camera to infer the objects dimensions. The main objective of this was to study if a better estimation of the objects occupancy along the HD map polyline can be obtained.

We have proposed to quantify the performance of a perception system under evaluation according to an integrity diagram for estimating the quality of objects occupancy. We observed that the LiDAR on-board system performs well, and in the particular configuration that we have studied, the information provided by the cooperative system does not significantly improve the LiDAR tracking system when the fields of view of the two systems overlap. This can be due to the fact of that.

The approach that we chose for updating the estimation of the occupancy intervals is quite pessimistic, i.e. it allows only to increase and not to decrease the size of the estimated occupancy. This means that the tracks tend to accumulate occupancy and grow, leading to a long term overly-estimated size.

In order to improve the proposed method, a possible perspective would be to improve the quality of the infrastructure information and to integrate it in the state estimation process of the detected objects in terms of pose and occupancy. Furthermore, an other approach would be to consider a map-aided tracker that tracks the bounds of the objects instead of considering the barycenter of perceived clusters. This would help to provide a more stable and consistent estimation of the state because, as we observed in our experiments, the shape of the detected objects can change a lot from one instant to another, with the consequence that the barycenter moves a lot too.

6 General Conclusions and Perspectives

The work presented in this manuscript addresses the problem of safe autonomous driving vehicle navigation in complex urban scenarios. This has been developed in the context of the French national project Tornado, which aimed at experimenting autonomous shuttles and robotaxis in urban driving environment. In the first part of work, we focused the attention on the navigation strategy and longitudinal planning in roundabouts in presence of a regular vehicle traffic flow. Then, in the second part of this research, we developed an on-board perception system to provide the self-driving vehicle information about the road users. Then, we proposed a cooperative data fusion strategy with a remote intelligent infrastructure to enhance the perception capabilities of the autonomous system and to provide a more robust and accurate estimation of the perceived road users.

6.1 High Definition Maps: Key Enablers for Autonomous Navigation

A contribution of this manuscript is the development of a longitudinal path planning method over a short time horizon. This method relies heavily on the High Definition (HD) map formalism to model the road environment. This formalism allows the autonomous vehicle to take decisions when interacting with the other road agents within the vehicle navigation corridor thanks to the HD map. One of the most important strengths of this approach is that it models the motion of the vehicle along the map polylines. This reduces the degrees of freedom needed to model motion of the vehicle. This is useful when performing a prediction over a time horizon of all the possible vehicles behaviors because it exploits the constraints provided by the map to reduce the number of possible configurations in the state space. For this reason, HD map polylines are useful to obtain a fast and sufficiently detailed formalism to find the possible interactions between the autonomous vehicle and the other road agents.

Furthermore, the use of a HD map can also be integrated to model the road agents dynamics. Indeed, by exploiting the topological information stored in the map, it is possible to extract a realistic path for the prediction of road agents behaviors. An example of that can be the use of speed limits information or the computation of an upper bound speed to reduce the set of all the possible outcomes over a time horizon.

Finally, a HD map is helpful also at the control level since it is possible to encode in the HD map some useful information that can be exploited by both the longitudinal and lateral controllers. In the former case, a speed profile can be encoded together with the polylines information to provide the autonomous vehicle a set-point speed to follow as a function of the map polylines. In the latter case, the HD map can be used as a reference path to be followed by the lateral controller.

On the flip side, even if HD maps offer an appealing solution for a wide set of autonomous driving tasks, they also have some limitations. Firstly, the HD map formalism works well to model driving situations that occur close to the road polylines. Map-matching has a tendency to place objects along the map polylines even if they are not close to the road center, as for example a parked car or some pedestrians along a sidewalk. To overcome this issue, a dedicated system able to select only the objects that belong to the driving scenario and that represent an obstacle for the self-driving car is needed. It remains challenging to develop a system that only selects the "true" road obstacles.

Secondly, one other difficult issue about HD maps is their maintenance. If one wants to use a HD map to perform safety-critical tasks, e.g. for navigation purposes, the information contained in the map has to be correct and representative of the real driving environment. Unfortunately, due to changes in the road environment, it is difficult to guarantee the consistency of such information over a long time horizon. Moreover, it is also challenging for the AD vehicle to detect with its own system an inconsistency on the driving environment representation.

A solution to partially resolve this could be to exploit the ETSI DENM (Decentralized Environmental Notification Message) to notify a self-driving vehicle about a modification in the environment. Such a notification should provide information on the nature of the notification, e.g. road works, accident, etc. and also a timestamp to notify the user.

Another alternative that it is gaining more and more attention nowadays is the crowd-sourcing of HD maps. This technique consists in providing to the HD map clients quasi real-time information about the modifications of the elements present in the map. For this to work properly, users are supposed to notify at any time changes and modifications on the map elements in order to enforce the correctness of the geographical information.

6.2 Autonomous Vehicles Navigation

Another contribution of this work is the design and the implementation of an algorithm for roundabout insertion maneuvers in urban traffic. This algorithm has been specifically designed to operate in a scenario where an autonomous vehicle navigates among a regular vehicle flow. One of the advantages of the proposed method is to use virtual instances of vehicles to predict uncertain intentions of self-driving cars both in path prediction and in multi-lane roundabout navigation.

This formalism allows to cover all the possible decisions about the uncertain intentions of MD vehicles thanks to the HD map-based representation and the concept of virtual vehicle. A limit of this approach is that it does not guarantee an optimal estimation of the driver's intention. To obtain such information, adding a layer in our architecture that is able to identify drivers intentions could improve the performance of the system. Moreover, this layer could be useful not only to identify for example the road that a car is going to take into an intersection, but also to identify the intention of the vehicles to do other maneuvers like lane change.

Another improvement that could be added is a negotiation layer to unblock a deadlock situation that may occur in high density traffic particularly for roundabout insertion. This can happen in conditions where, due to traffic jams, human drivers tend not to respect traffic rules and, for example, take the right of way even if they do not have the right. All the roundabouts are priority-based. It means that vehicles in the roundabout ring have the right of way w.r.t. incoming ones. In the

proposed strategy, this priority constraint has been handled when there is a large enough gap between vehicles on a priority branch. In other words, a vehicle that does not have the right of way can effectuate an insertion maneuver if and only if it does not influence the behavior of another vehicle with a higher priority rank.

Another contribution of work part is the extension of the single-lane roundabout crossing algorithm to multi-lane ones. This extension allows to handle more complex traffic scenarios and to choose among three policies to effectuate the roundabout insertion maneuver. Again, we have chosen to use the most pessimistic one to ensure that, under no circumstances, collisions due to the wrongly estimated intention of a manually driven vehicle can occur. It is also possible in this case to add a layer to estimate intentions of changing lane to improve the overall number of insertions.

Another future research direction on this topic is to take into account the lack of information about the surrounding driving scenario. Indeed, the current perception system may contain some zones that are occluded by the presence of road obstacles. The presence of large objects as trucks or vans may hide the presence of another obstacle behind them. This means that such an obstacle is out of the on-board system field of view and cannot be perceived. However, this obstacle can potentially represent a danger during the insertion maneuver. To avoid such an issue, the proposed strategy needs to be reinforced with a system capable of identifying the occluded zones and to include this notion in the planning step during safety critical maneuvers.

Finally, the use of a more sophisticated control strategy (as for example the model predictive control) would allow to compute reference signals over a time horizon, providing smoother signals and embedding some navigation constraints directly in the control layer.

6.3 LiDAR-based Road Users Localization

Road users detection and localization with 3D LiDAR is a point that has been considered carefully in this research. Such a system has been used to provide the AD vehicle the necessary knowledge of the road users as vehicles. We have implemented a multi-layer approach that processes raw LiDAR point clouds data and identifies the main road obstacles using geometric methods. We have chosen to use geometry-based techniques in contrast to machine learning based ones because it is well known that those methods provide fewer missed detections w.r.t. the other category. This is because there is no need to provide a database of potential obstacles to train the algorithm. Indeed, machine learning is only able to recognize objects correctly that belong to the training set. This may lead to dangerous situations when an unknown object appears.

Conversely, even if geometry-based methods are more reliable in terms of missed detection rate, they have strong limitations for estimating a consistent occupancy of the perceived objects. They lack prior knowledge that can infer important information about perceived objects like for instance the nominal objects dimension. Adding a classification layer, as we did in the case of the intelligent infrastructure perception system, could help in better estimating the objects dimension.

To meet the needs of our navigation strategy, the perception system must be able to provide the estimated occupancy of road obstacles. In this work, we chose to estimate the surface occupied by road users from the objects detected by the LiDAR clusters. There is a dual approach. Instead of estimating the occupied space starting from the LiDAR obstacles, it is also possible to perform such an estimation

by considering the free space instead of the occupied one. This information can be retrieved from the ground points outputted from the first block of the pipeline. This point cloud contains all the points that have been classified as ground and they can be used to infer the zones of the drivable space that are free of obstacles. Following this reasoning, one can obtain the information about the occupancy of road obstacles by considering the intersection between the estimated free space and the drivable space. One can also refine this approach by adding more sophisticated models that consider for example the unknown space to model cluttered environments or free space that has not been perceived by the on-board perception. This can be obtained by adding a free space detector based on the ground segmentation module to the presented pipeline.

Regarding the integrity of perceived bounding polygons with respect to the AD vehicle localization uncertainty and the map-based object filtering, we have demonstrated experimentally that the proposed method is safe. The pessimism it generates should be assessed more closely.

Furthermore, this method can also be useful if one is interested in estimating the effects that different sources of localization can produce on perception results. For instance, the AD vehicle localization uncertainty depends on the quality of the GNSS receiver. It is possible to test how the localization uncertainty provided by different GNSS technologies is propagated to the perception results and which is its impact on the resulting bounding polygons in terms of integrity and occupancy of the lanes. This is useful to evaluate the percentage of space occupied by MD vehicles and in function of that, to choose which GNSS technology is better to use to obtain a trade-off between localization accuracy and space occupancy for the AD vehicle navigation tasks.

Finally, in the experimental part, we have compared the LiDAR perception system in the curvilinear map-based coordinates with the Cartesian framework. From the point of view of integrity, we found out that working in a curvilinear framework provides better performance than in Cartesian coordinates. For safety-critical applications, we therefore recommend this formalism to obtain a consistent representation of the driving environment, especially if a map-based filtering step is involved.

6.4 Curvilinear Tracking

The tracking of the perceived objects constitutes an important step to provide to the navigation algorithm the necessary information to run. In this work, a tracking step has been added to estimate perceived objects speed and occupancy. We have implemented a map-aided algorithm for objects tracking. The main motivation of this choice was to provide the tracking result along the HD map polylines. This is coherent with both the representation of the driving environment and the formalism adopted to design the navigation strategy. However, tracking objects w.r.t. a curvilinear framework may lead to poor performance when the detected objects do not follow the driving behavior encoded in the HD map. As a consequence, the tracking system is not able to encode all the possible driving behaviors.

Working in curvilinear coordinates allows the system to compute occupancy intervals directly in the curvilinear framework and with respect to the navigation corridors provided by the map. This formalism allows to compute the occupancy only on the along direction of the motion and avoids the computation of oriented 2D bounding boxes.We leave as a future perspective the comparison of the performance of this system with a general 2D Cartesian tracking algorithm.

During the experiments, we observed that considering the barycenter of the perceived objects as the observation for the tracking algorithm degrades the performance. This is because the position of this point heavily depends on the shape of the perceived cluster and, considering LiDAR-based perception, it can vary significantly over time. This depends on the relative position between the sensor and the tracked object. An alternative could be to change the tracking approach by replacing the barycenter tracking with the tracking of the occupancy interval boundaries. Such information can be helpful not only to better stabilize the tracked object estimation, but also at the navigation layer to quantify the uncertainty about the boundaries of the occupancy interval of a perceived object.

Finally, another improvement to our approach could be to choose a wiser policy for updating the size of the estimated occupancy interval. One could develop a criterion that allows to include outliers detection in order to avoid to overly estimate objects sizes. This could be helpful to improve the pessimistic estimation of objects occupancy without introducing risks w.r.t. the integrity criterion.

6.5 Cooperative Perception with Intelligent Infrastructure

Another aspect that we considered in this work was how an extra perception coming from a remote intelligent infrastructure can help the autonomous vehicle to improve its knowledge of the driving environment. Firstly, the different positions of the two sensors allow to extend the field of view of the AD vehicle and the scene is perceived from different directions. This aspect is important to detect road users sufficiently ahead. As a consequence, the autonomous vehicle can decide to use more elaborated navigation algorithms and to plan its navigation strategy with a clearer knowledge of the driving environment. This aspect was shown in our experimental tests for roundabout crossings, where we compared the gain of the cooperative system again the standalone ones. Secondly, the remote infrastructure also offers an appealing solution to robustify the knowledge of perceived objects by using the infrastructure observations in the LiDAR-based objects tracking of the self-driving car to improve the estimation of objects occupancy via a data fusion technique. To do so, it is necessary to synchronize temporally both the remote and the on-board systems.

Moreover, we have shown that a multi-sensor data fusion technique not only helps in enhancing the AD vehicle on-board perception field of view, but also it allows to better estimate the state of perceived objects in situations where the two sensors fields of view overlap. This brings to light one discussion about the positioning of the infrastructure in the cooperative system either to optimize the complementarity of the fields of view or to maximize the overlapping zone in order to provide a more robust state estimation of perceived objects. Because a monocular camera was used, using the objects detected from the image frame to estimate their occupancy on the road plane in the world frame introduced a additional layer of uncertainty and inaccuracy on the final result.

Another aspect to obtain a good performance in real-time is the optimization of all the delays accumulated during the infrastructure processing of the data. To work well, this approach needs to have the minimum delay between the moment when a raw image is acquired and the moment when the detection result is received by the data fusion framework. Between these two points, there are some steps as the transit of the data in the system, the image treatment, the encoding of the detection

result into a CPM message, etc. If one of these sub-step is not optimized, it might happen that the data are received by the autonomous vehicle with too long a delay.

Finally, considering the data-fusion technique used in the cooperative system, we found out that it provides only a negligible improvement in terms of integrity of occupancy estimation considering the results provided by the LiDAR-only system. However, due to the large quantity of uncertainty accumulated during the occupancy update and the criterion that we chose to update tracked objects occupancy, it could be helpful to exploit the classification information and the inferred objects dimensions provided by the infrastructure to fix an upper bound on objects occupancy. This implies that object occupancy will not be allowed to grow more than their real lengths. However, if we are in the case where it is not clear if the barycenter of the tracked occupancy corresponds or not to the real object one, we suggest majoring the occupancy estimation with the double of the inferred length, in order to provide a further upper bound. This approach results again in pessimism but it has the advantage that it always contains the perceived object.

6.6 General Perspectives

This work has highlighted the importance of using HD maps for autonomous vehicles navigation and to exploit them for safety-critical navigation tasks. Furthermore, the emerging concept of integrity of perceived environment has been addressed at several levels. Integrity of perceived environment represents a crucial aspect in several tasks of autonomous driving. However, some issues still remain. Regarding autonomous driving use-cases, the results provided by applying the integrity criteria to perception add unavoidably some pessimism. This is mainly due to the uncertainty inclusion in the computation of the occupancy estimation. As a consequence, autonomous driving algorithms tends to be overly-cautious.

To overcome this issue, it is required to find a trade-off between the integrity criterion and an excessive pessimism of the system in terms of overly-estimated occupancy. One practical solution could be to exploit the classification information about perceived objects, coming from either the intelligent infrastructure or from an on-board camera system, to add more constraints to reduce the final amount of occupied road surface by the perceived objects. This has the goal of making the estimated occupancy as close as possible to the real one.

In a more general way, exploiting information from different sensors with different modalities can help in providing different pieces of information about the driving environment. When such information is puzzled out, it can give a more detailed, robust and accurate description of the perceived objects.

Another aspect that we would like to point out concerns cooperative perception systems. In the presented approach, we exploited a remote cooperative perception source to enhance and extend the self-driving vehicle knowledge of the driving environment. However, this domain offers a plethora of different solutions to cooperative perception problems. A crucial aspect of this is the format that data should have in order to be proficiently exchanged between road users. On the one hand, cooperative systems tend to reduce the amount of information to be send in order to minimize latency and to avoid network saturation. This is often implemented by providing a more compact and object oriented representation of the perceived information. On the other hand, raw data are more informative w.r.t. treated ones because there is no information loss during the sharing process. Nonetheless, it could be challenging to send raw data because, for certain sensors, they require a huge amount of memory

to be stored.

Furthermore, in the field of cooperative systems it is also important to obtain redundant information coming from sensors that use different perception modalities from different points of view. In our specific case, we exploited a 360° LiDAR and an intelligent fixed camera. It is useful to consider the cooperative perception problem as a function of the different sensors involved with their own fields of view. For on-board perception systems, it would be interesting to investigate the infrastructure benefits in the case of on-board systems that do not have a 360° vision range, like a forward looking camera for instance. In such a case, it would be useful to perform cooperative multi-sensor data fusion with both on-board and remote sensors that scan the world from different positions and directions. It can be really helpful to improve the overall system performance by increasing the coverage of the scene perceived and reducing occlusions.

Regarding the intelligent infrastructure, it would also be interesting to evaluate the benefits of a remote system that exploits a technology different from the one of vision cameras. Some other interesting alternatives could be to explore infrastructure LiDAR sensors and radars to provide a better estimation of the speed of the detected objects.

Cooperative systems with remote infrastructure offer a trustworthy additional source of information to cooperative perception. The remote position of the infrastructure helps in providing a different vision of the driving situation from another perspective and its contribution offers, in general, an enhancement of the cooperative perception system field of view. To proficiently exploit the information provided by the infrastructure at the data-fusion level, an optimization process to minimize latency and delay due to information broadcast is necessary to obtain a sufficiently high refresh rate that meets safety criteria for autonomous vehicles.

To summarize, we can state that an intelligent infrastructure helps in enhancing the field of view of autonomous vehicles on-board perception which is very useful for the anticipation of the driving situation and therefore at the tactical level of navigation. Because of the processing and broadcast delays, the infrastructure is not sufficiently adapted to assist reactive problems with strong real-time constraints. In practical implementations, the cost of equipping a roundabout with such an extended perception system is maybe not proportional to the benefits it provides. However, some practical use-cases where this technology can provide a significant benefit are, for example, private sites as warehouses or parking lots or, regarding public roads, some particular traffic zones where the risk of road accident is high like roundabouts with blind spots or occluded zones.

7 Appendices

7.1 ETSI Standards for Vehicles Communication

In this first appendix, we provide a detailed explanation of the main standards used to achieve vehicular communication in the autonomous vehicles domain. In particular, we focus the attention on the European standard ETSI for vehicles communication. In the chapter, we detailed the main types of messages contained in this standard and we also provide a brief comparison with the American SAE standard. Moreover, we discuss in details the format and the contents of the Cooperative Perception Message (CPM) providing an evaluation about the relevance of its different fields to our I2V communication use-case.

7.1.1 Introduction

Cooperation in autonomous driving is an emerging technology that is looking at to enhancing the performance of autonomous vehicles by means of cooperation in Intelligent Transportation Systems (ITS). In order to implement cooperation, it is required to share information between road users. Vehicular communication is one of the most important means to share such information among several users. Moreover, the communication can be implemented not only between road users, but also between road users and Road Side Units (RSU) in both directions. The aim of this section is to present an overview about of the existing standards used in vehicular communications. Furthermore, we analyze also the application of such communication standards to ensure the exchange of information with an intelligent infrastructure in order to ensure safe navigation. In particular, we focus on the necessary information and its format that RSUs and self-driving cars have to exchange to achieve enhanced perception.

There exists in the literature a wide range of scenarios that take advantage of inter-vehicles communication. All these cases can be grouped into three principal subsets:

- **Safety-oriented**: These applications aim to ensure driving safety. This imposes many constraints on the communication standards as for example real time constraints, low message latency and low messages loss rates. In general, the aim of these applications is to make cars exchange data about their current status, their perceived environment and the notification of a perturbing event. Exchanging such information between cars in a road network can provide to single cars a global and enhanced view of the surrounding environment that a driver in general cannot have. Some example of that are the possibility of seeing cars in bad weather conditions, the advertising of traffic jams and car accidents on the road, enhancing a vehicle perception on cluttered environments and blind spots and the advertising of the presence of an incoming emergency vehicle.

- **Traffic-Control Oriented**: These applications are not related to safety. However, exchanging information about the vehicles positions can help in individuating the zones with traffic congestion giving a global view about the traffic. This can be used to regulate the traffic in function of the traffic jams. A typical use case is a smart traffic light manager that collects information about the queues of cars that are waiting to pass and regulates the passage minimizing the waiting time.

- **User-Comfort Oriented**: These applications are oriented at providing services that a user can enjoy while he is driving on the car. Some examples are the possibility of downloading movies or music during a trip. All these cases have the basic requirement of having access to the internet.

7.1.2 ETSI Standard

In Europe, the first experiments to achieve vehicular communication started around 1980. Before that date, several projects had been launched to achieve cooperation among communicating devices. The standards to achieve these tasks have been developed by the European Standardization Union (ESOs), the European Telecommunication Standards Institute (ETSI) and the Comité Européen de Normalisation (CEN). This standardization covers all types of transportation systems and also infrastructure based systems as the tolling systems. The standardization is driven by the Car-2-Car Communication Consortium (C2C-CC), which is an industry consortium of automobile manufacturers that signed an agreement to introduce the standard in Europe since 2015.

The ETSI ITS standard is based on the concept of ITS station. An ITS station can be a connected vehicle, a person with a connected device, or a communicating roadside unit. In the United States, another standard called WAVE has been developed. Figure 7.1 shows the architecture of the ETSI standard compared with the WAVE architecture. The access technologies layer primarily utilizes a specific set of options of the IEEE 802.11 standard, that is, ITS-G5 (where G5 stands for 5 GHz). In the United States, this set is named Wireless Access in Vehicular Environment (WAVE), formerly referred to as the IEEE 802.11p amendment and now integrated into the IEEE 802.11-2012 standard release. The European variant, ITS-G5, is derived from WAVE and adapted to European requirements.

On the top of the network and transport layer, there are the standards for application-oriented vehicular communication. Among these facilities, there are two services for cooperative communication. The CAM protocol conveys critical vehicle state information in support of safety and traffic efficiency application. This is useful because receiving vehicles can track other vehicles positions and movement. The DENM protocol disseminates event-driven safety information in a geographical region.

Finally, the new Cooperative Perception Message (CPM) protocol (which is not present in Figure 7.1 because it has been recently introduced) allows vehicles to exchange information about their perceived environment. This is useful to enhance onboard sensors' fields of view and to obtain a extended and more consistent representation of the driving environment. It will be presented below.

From Figure 7.1, one can see that the upper layers facilities are implemented by the CAM and DENM (and CPM) protocols for the ITS-G5 standard, while for the WAVE standard there are the BSM and SAE. Regarding the lowest layers, we can say that the standards are very similar in both versions.

II. ITS-G5 IVC STANDARDIZATION

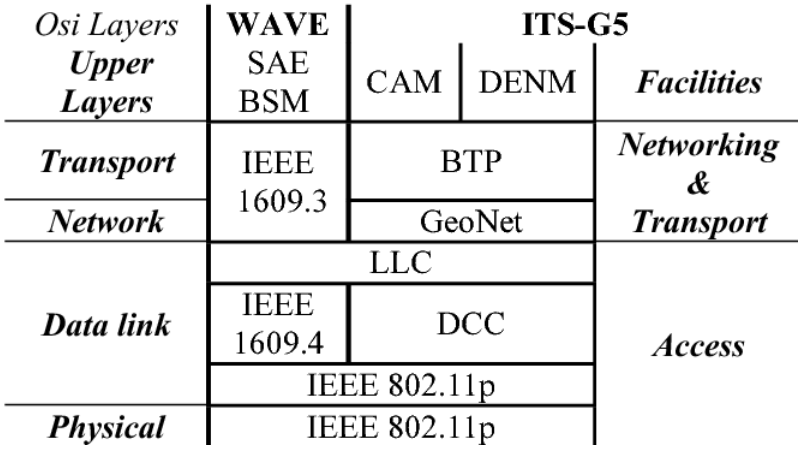


Figure 7.1: Architecture of ITS ETSI standard compared with the WAVE architecture. Figure from [1]

Let us briefly show the contents of the SAE standard. We do not provide details about the lower level architecture of the system, because we are interested in analyzing and comparing only the services provided by higher level application levels. In particular, we are interested in the SAE J2735 standard that specifies the dictionary for the Base Safety Message (BSM) which is used to achieve the main tasks about navigation safety. Figure 7.2 shows the messages list available on SAE J2735 standard, while Figure 7.3 displays the contents of a BSM. It is easy to see that there exist similarities between the European and the American Standards. A brief explanation of the most important messages reported in Figure 7.2 is given hereafter:

- **Basic Safety message**: a message that is constantly exchanged between the neighboring vehicles to inform all the other ITS about the presence of a vehicle.
- **A la carte message**: a message customizable by the user which allows flexibility in data representation.
- **Emergency vehicle alert message**: used used to broadcast warnings to the surrounding vehicles of an emergency vehicle operating in the neighborhood.
- **Generic transfer message**: a basic mean to exchange data between a vehicle and the roadside unit.
- **Common safety request message**: used when a vehicle that exchanges BSM needs to make specific requests to other vehicles for additional information about safety applications.

7.1.3 Cooperative Awareness Messages (CAM)

To provide cooperative awareness between several communicating road entities, a Cooperative Awareness Service has been implemented. The standard is defined into the ETSI EN_30263702 document [1]. This standard can be applied to every road vehicle (cars, motorbikes, truck, etc.. . .), to pedestrians and to roadside infrastructure units (intelligent traffic lights, barriers, tolls, etc...). To implement cooperative awareness, the information to be shared is exchanged by means of the Cooperative

J2735 Defined Messages

ID	Messages	Typical Use	Status
0	Reserved	N/A	
1	MSG_A_la_Carte	V2X	
2	MSG_BasicSafetyMessage (BSM)	V2V	Used by USDOT program & other ITS industry research
3	MSG_CommonSafetyRequest	V2?	
4	MSG_EmergencyVehicleAlert		
5	MSG_IntersectionCollisionAvoidance	V2X	
6	MSG_MapData	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
7	MSG_NMEA_Corrections	I2V	
8	MSG_ProbeDataManagement	I2V	Used by VII Proof of Concept (PoC) project
9	MSG_ProbeVehicleData	V2I	Used by VII PoC project
10	MSG_RoadSideAlert		
11	MSG_RTCM_Corrections	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
12	MSG_SignalPhaseAndTiming	I2V	Based on USDOT/CAMP CICAS-V project. Used by various demo/research program
13	MSG_SignalRequestMessage	V2I	
14	MSG_SignalStatusMessage	I2V	
15	MSG_TravelerInformation Message	I2V	Used by VII PoC & will be used in Model Deployment (Curve Speed Warning)

Figure 7.2: A list of different messages contained into the J2735 standard. Notice that several messages corresponds to the same use-cases of ETSI. Figure taken from [2].

Data elements/frames		Description	Remarks
Part I	DSRCmsgID		
Part I: BSM Blob (Octet string)	MsgCnt		
	TemporatyID		
	DSecond		
	Latitude		
	Longitude		
	Elevation		
	PositionalAccuracy		
	TransmissionAndSpeed		
	AccelerationSet4Way		
	BrakeSystemStatus		
	VehicleSize		
Part II	SafetyExtension		Optional
	VehicleStatus		Optional

Figure 7.3: The payload of a BSM message. Figure taken from [2].

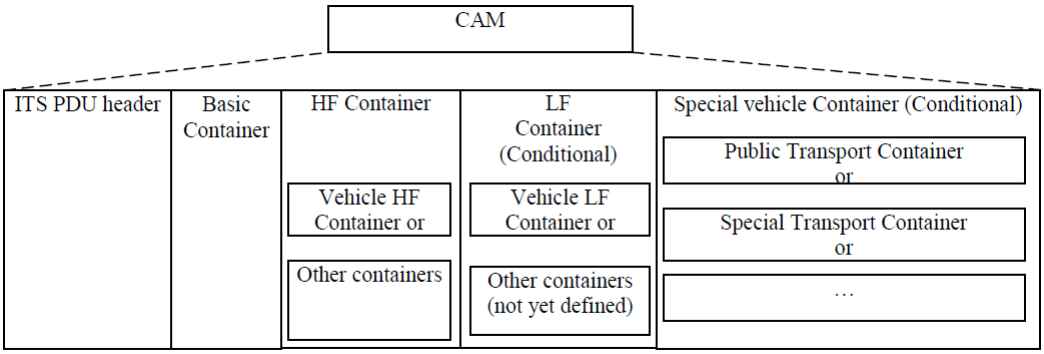


Figure 7.4: The format of an ETSI CAM message. Figure taken from [1]

Awareness Message (CAM). To implement this, every road users has to exchange information about its status. On reception of a CAM, the receiving user becomes aware of the existence of the sending one and its current status. Several use cases for this service are present in literature, as it is explained in [1].

7.1.3.1 CAM Generation Frequency

To implement the cooperative awareness service, it is important to update periodically the status of the surrounding environment. For safety critical scenarios, the evolution of the driving situation is highly dynamic. To achieve this, the CAM standard proposes some constraints:

- The CAM generation interval should not be inferior to 100 ms (10 Hz).
- The CAM generation interval should not be superior to 1000 ms (1 Hz).

The upper bound limit of 10 Hz has been chosen to avoid network saturation and bottlenecks that can be caused by multiple sending of several road agents.

7.1.3.2 CAM Messages Format

CAM is composed by a "ITS-PDU" header and several containers, which together constitutes a CAM. The ITS PDU header is a common header that includes the information of the protocol version, the message type and the user ID of the originating user. Figure 7.4 illustrates the structure of a CAM.

To successfully implement cooperative awareness, a CAM must include at least one basic container and one high frequency container. It can also have one low frequency container, and one or more special vehicle container:

- The basic container includes basic information related to the originating user.
- The high frequency container contains highly dynamic information of the originating user.
- The low frequency container contains static and not highly dynamic information of the originating user.
- The special vehicle container contains information specific to the vehicle role of the originating vehicle user.

7.1.3.3 CAM Payload Description

Let us see the information necessary to implement the CAM basic services. In this part, we decided also to assign a color to each field of the payload. Such colors are assigned to fields according to the criteria explained in Table 7.1.3.3. Colors are assigned to each field in order to specify the importance of every piece of information in the context of a safety-critical autonomous vehicles navigation application. The following table 7.1.3.3 explains the meanings of each color.

Red	High priority information	Used to highlight information that is necessary to develop an application for road safety.
Yellow	Middle priority information	Used to have some information that is not considered necessary to implement safety-oriented applications. However, it can be useful to have its knowledge.
Green	Low priority information	This information is not considered relevant for our use-case. It can be eventually erased to make space for other more relevant fields.
Blue	High priority information	Used to highlight information that is necessary to develop an application for road safety. However, this information is, in general, hard to compute
Classification criteria for messages fields w.r.t. our use-case application.		

Basic Container This container provides basic information about the identity of the user. The following table 7.1.3.3 explains in detail the contents of this container.

Importance	Field Name	Description
R	Station Type	Station type of the originating user.
R	Reference Position	Position and position accuracy measured at the reference point of the originating user. The measurement time shall correspond to generationDeltaTime. The positionConfidenceEllipse provides the accuracy of the measured position with the 95 % confidence level. Otherwise, the positionConfidenceEllipse shall be set to unavailable.
R	Timestamp	Time corresponding to the time of the reference position in the CAM, considered as time of the CAM generation.
Basic Container		

High Frequency Container This container provides highly dynamic information about a certain user. This information must be refreshed frequently in order to keep the state updated. Table 7.1 illustrates the container's field in details.

Importance	Field Name	Description

B	Heading	Heading and heading accuracy of the vehicle movement of the originating user with regards to the true north. The heading accuracy provided in the DE headingConfidence value shall provide the accuracy of the measured vehicle heading with a confidence level of 95 %. Otherwise, the value of the headingConfidence shall be set to unavailable.
R	Speed	Driving speed and speed accuracy of the originating user. The speed accuracy provided in the DE speedConfidence shall provide the accuracy of the speed value with a confidence level of 95 %. Otherwise, the speedConfidence shall be set to unavailable.
R	Driving Direction	Vehicle drive direction (forward or backward) of the originating user.
Y	Vehicle Length	This DF includes: • vehicleLengthValue: Vehicle length of the vehicle user that originates the CAM. If there are vehicle attachments like a trailer, or overhanging attachments like a crane, that extend the vehicle length to the front and/or rear; then the vehicleLengthValue shall provide the length for the vehicle including the attachments. • vehicleLengthConfidenceIndication: indication of whether trailer is detected to be present and whether the length of the trailer is known.
Y	Vehicle Width	Vehicle width, measured of the vehicle user that originates the CAM, including side mirrors.
R	Longitudinal Acceleration	Vehicle longitudinal acceleration of the originating user in the center of the mass of the empty vehicle. It shall include the measured vehicle longitudinal acceleration and its accuracy value with the confidence level of 95 %. Otherwise, the longitudinalAccelerationConfidence shall be set to unavailable.

G	Curvature	This DF is related to the actual trajectory of the vehicle. It includes: • curvatureValue denoted as inverse of the vehicle current curve radius and the turning direction of the curve with regards to the driving direction of the vehicle • curvatureConfidence denoted as the accuracy of the provided curvatureValue for a confidence level of 95 %.
G	Curvature Calculation Mode	Flag indicating whether vehicle yaw-rate is used in the calculation of the curvature of the vehicle user that originates the CAM.
R	Lane Position	The DE lanePosition of the referencePosition of a vehicle, counted from the outside border of the road, in the direction of the traffic flow. This DE shall be present if the data is available at the originating user. This concept can be computed in a curvilinear framework if a map is available.
B	Steering Wheel angle	This DF includes the steering wheel angle and accuracy as measured at the vehicle user that originates the CAM. It consists of the following DEs: • steeringWheelAngleValue denotes steering wheel angle as measured at the vehicle user that originates the CAM. • steeringWheelAngleConfidence denotes the accuracy of the measured steeringWheelAngleValue for a confidence level of 95 %. Otherwise, the value of steeringWheelAngleValue shall be set to unavailable.

G	Lateral Acceleration	Vehicle lateral acceleration of the originating user in the center of the mass of the empty vehicle. It shall include the measured vehicle lateral acceleration and its accuracy value with the confidence level of 95 %. This DE shall be present if the data is available at the originating user.
G	Vertical Acceleration	Vertical Acceleration of the originating user in the center of the mass of the empty vehicle. This DE shall be present if the data is available at the originating user.
G	Performance Class	The DE performanceClass characterizes the maximum age of the CAM data elements with regard to the generationDeltaTime
G	CenDRSCTollingZone	
G	Yaw Rate	This DF includes: • yawRateValue denotes the vehicle rotation around the center of mass of the empty vehicle. The leading sign denotes the direction of rotation. The value is negative if the motion is clockwise when viewing from the top. • yawRateConfidence denotes the accuracy for the 95 % confidence level for the measured yawRateValue. Otherwise, the value of yawRateConfidence shall be set to unavailable.
G	Vehicle Role	The role of the vehicle user that originates the CAM. The setting rules for this value are out of the scope of the present document.
G	Vehicle Light	Status of the most important exterior light switches of the vehicle user that originates the CAM.
G	Path History	This DF represents the vehicle's recent movement over some past time and/or distance. It consists of a list of path points, each represented as DF PathPoint. The list of path points may consist of up to 23 elements.
High Frequency Container		

Special Container This container must be filled with another container according to the type of special vehicle and its characteristics we want to describe (e.g. if it is a police car, it is communicated whether the car is on an emergency status or not). This part is not considered relevant to our study and so it is not described in detail.

7.1.4 Cooperative Perception Message (CPM)

The main goal of a Cooperative Perception (CP) basis service is to share with others road users the environment perceived by a vehicle with its own sensors. The perceived environment representation can be refined, fused, processed and classified before the broadcast. Final results are stored into CP objects and broadcast to other road users. Moreover, some quality indexes can be added to perceived objects, in order to quantify the information reliability and consistency.

A CP object contains an aggregated and interpreted abstract information perceived by sensors about other road participants and obstacles. Typically objects are represented in a mathematical formalism i.e. a set of variables describing characteristics as their dynamics, their geometry and several other aspects.

In this part, we are also interested in analyzing which information is relevant to exchange perception between several road users. In particular, we focus our attention to safety-critical use-cases and we investigate the required level of information to proficiently ensure safety. Obviously, a trade off between detailed information and payload size of the message exists. Such constraint implies that we must investigate the essential information to be shared among road users in order to both provide a compact and complete environment representation and to avoid sending redundant information.

Finally, the object representation is sent to other surrounding vehicles exploiting the V2X communication. With this new received knowledge, other users can enhance their environment representation and complete their knowledge of the ongoing road scenario with information that is not directly accessible.

Some examples where this service can improve the performance are the filling of blind spots and cluttered environments. In our specific case, we exploit a remote intelligent infrastructure to provide the autonomous vehicle additional sources of information via I2V communication. For this reason, in the following part, we investigate the contents of a CPM message to select information that is necessary to broadcast to ensure safe navigation in a roundabout.

7.1.4.1 Cooperative Perception versus Cooperative Awareness

Cooperative Perception is the concept of sharing perceived environment of a road user to others. This perception is based on information obtained from sensors. The main difference between cooperative perception (CP) and the cooperative awareness (CA) is that, in the first case, the broadcast information is about the vehicle current environment, rather than about the vehicle current status. However, it is mandatory to include in a CP basic service information about the sending vehicle in order to reference the perceived objects in other vehicles frames.

7.1.4.2 Messages Transmission and Generation

The road user is supposed to send a CPM whenever it detects an object with a sufficient level of confidence. It is possible to not send a detected object because the confidence level on the detection is low. However, even if the object is rejected,

the user should send a CPM (at least empty, if no objects are reputed to have good confidence) at minimum sending frequency. The empty container must have inside it the information about the sending vehicle. This can help other road users in knowing the following things:

- A road user is present on the scenario
- A potential additional source of information is present on the scenario

Transmission rate of a CPM is computed according to the following criterion:

- Broadcast information should be as detailed as possible and provided as frequently as possible
- The utilization of the channel should be minimized

According to this, the CP basic services define the limits of the interval between two consecutive CPMs (and the corresponding sending frequencies) as follows:

- $T_{genCpm} > T_{genCpmMin}$, with $T_{genCpmMin} = 200$ ms (CPM generation rate of 5 Hz)
- $T_{genCpm} < T_{genCpmMax}$, with $T_{genCpmMax} = 1000$ ms(CPM generation rate of 1 Hz)

A CPM message needs to be generated when:

- A new object is detected
- A change in position of a previously declared static object is detected
- A change in velocity or position of a previously declared dynamic object is detected.

7.1.4.3 Object Confidence

Transmitted data should be as close as possible to the original data. In such situation, one idea could be to send directly sensors raw data. This has the advantage that no information loss occurs. In facts, perceived information is transmitted in its raw data form without introducing any kind of treatment on data. However, it is not possible to send directly raw data because it is not feasible in terms of channel load and there is no guaranty that the receiving user has the necessary means to process them. For this reason, it is often preferred to send a compact representation of the perceived environment. If the channel constraints allow it, it is also possible to attach into the CPM some raw data representation.

In order to fulfill the whole requirements of CPM standard, some processing at low level is needed. In particular, it is required to send an object and to provide an index that quantifies the confidence level of the provided information. This has to be done to provide the receiving user a mean to evaluate the quality of a detected piece of information. Such confidence needs to be computed in a way that it can have the same meaning for every user that has access to the shared object. However, there exist some cases in the literature where this value depends on the particular method that has been used to compute it. Confidence needs to be computed considering coherency with previously sent CAMs. This can help in tracking object and in associating the new detection to the previously existing ones.

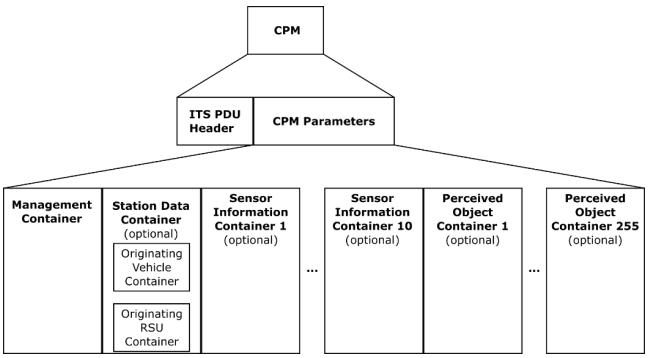


Figure 7.5: General structure of a CPM message. Figure courtesy of [1]

7.1.4.4 Objects Localization and HD Maps

CPM deals only with objects that move or have the ability to move. This assumption correspond to driving scenarios. This imposes that objects must be located on the driving lanes or pedestrian walks. If a map-based representation of the driving environment is available, it can be useful to have map matching procedure to localize objects on the scene at lane-level. Such phase has to be included into the CPM pre-processing part. Moreover, the map matching results should be sent to other users, in order to provide also the vehicle location inside a high-definition (HD) map and also their occupancy at lane level. However, to exploit such information in a proficient way, one needs to assume that the HD map is the same for every agent, which is, in general, not true. In our case, we assume that both the remote intelligent infrastructure and the autonomous vehicle share the same HD map representation of the driving environment.

7.1.4.5 CPM Message Format

The CPM format is made of several containers, as the usual structure of ETSI standard messages. In Figure 7.5, we illustrate the general structure of a CPM message.

On Figure 7.5, the ID of the sender is contained into the ITS PDU header. As we did for the CAM before, we need to ensure that every sender (Vehicles and RSU) has a unique identifier. This time, we also need to specify if the sender is a vehicle or a road side unit (RSU).

If the sending entity is a vehicle, it is strongly advised to specify into the Originating Vehicle Container the information about the dynamic of the vehicle (if it is available). On the other hand, if the message is generated by an RSU, containers need to provide references to identify the infrastructure on the HD road map. This is useful to localize information in the correct working frame. In our work, we consider the map frame as the world frame. As a consequence, information perceived by both the intelligent infrastructure and the autonomous vehicle onboard sensors is converted into such frame to be taken into account during navigation.

7.1.4.6 CPM Payload Description

As we did previously, we have decided to assign a color to each field of the payload. Such colors are assigned to fields according to the criteria explained in Table **??**. Colors are assigned to each field in order to specify the importance of every piece of information in the context of a safety-critical autonomous vehicles navigation application. Contrary to the scenario taken into account in section 7.1.3.3, we

consider now a use-case where there is only one AD vehicle in a driving scenario with only MD vehicles. Direct communication exists only between the infrastructure and the AD vehicle and, of course, information about other road agents needs to be estimated by the AD vehicle perception system. Table **??** explains the meaning of each color.

Importance	Field Name	Description
R	Station Identifier	It allows to identify the source of a certain CAM. It also permits to distinguish between several sources of information.
Y	Station Type	Tells us if the sending user is a vehicle or an infrastructure (RSU)
R	Reference position	It provides a position to reference perceived objects relatively to a global provided position. Detected objects are referenced into the vehicle's body frame. Once a CPM is shared, the receiving user should be capable of converting received data in their own frames.
R	Timestamp	A timestamp that indicates the time at which the cam has been sent by the user. It is important to distinguish between the sending timestamp of a CAM message and a timestamp used to date perceived objects.
Management Container Information.		

Importance	Field Name	Description
B	Heading	Value of the vehicle's heading w.r.t. to the true north with a 95% confidence level. This data can help in knowing the vehicle intentions in terms of trajectory. We need to clarify the difference between vehicle heading and vehicle orientation.
R	Speed	Driving speed of the sending vehicle. This measure should be provided with a 95% confidence level.
B	Vehicle Orientation	Angle and angle accuracy of the disseminating vehicle absolute orientation. This value is not equal to the heading, that is computed considering the speed value. An accuracy with a confidence level of 95% should be provided.
R	Driving Direction	Vehicle driving direction (Forward or Backward)
R	Longitudinal Acceleration	Vehicle longitudinal acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Lateral Acceleration	Vehicle lateral acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Vertical Acceleration	Vehicle vertical acceleration of the originating user at the reference point of the vehicle. Accuracy value with the confidence level of 95% should be included.
R	Yaw Rate	Rotation of the vehicle around its center of mass with its 95% confidence level
R	Path	The nominal trajectory of a vehicle. In a map-based approach, Path is represented as an ordered list of identifiers of the road links.
R	Pitch Angle	Vehicle pitch angle with 95% confidence level.
R	Roll Angle	Vehicle roll angle with 95% confidence level.
R	Vehicle Width	Width of the sending vehicle with 95% confidence level
R	Vehicle Length	Length of the sending vehicle with 95% confidence level
R	Vehicle Height	Height of the sending vehicle with 95% confidence level
R	Trailer Details	Detailson eventual vehicle trails
Originating Vehicle container information		

Importance	Field Name	Description
R	Intersection ID	Allows to link a CPM perceived from a given intersection to an existing intersection on the HD road map
Originating RSU Container Information		

Importance	Field Name	Description
R	Sensor ID	An identifier of a sensor. This pseudonym is used to relate sensor measurements to the sensor that perceived the measurements. A correspondence between the perceived objects and the corresponding sensor id should be instantiated.
R	Sensor type	Type of sensor. (Enumerated value). This field can indicate information not only from a single sensor, but also information fused from several sensors.
R	Vehicle Sensor	Specifies if the sensor is mounted on a vehicle, other characteristics are provided in [tab vehicle sensor]
R	Stationary Sensor	Specifies if the sensor is mounted on a roadside infrastructure, other characteristics are provided in [tab infrastructure]
Sensor Information Container		

Importance	Field Name	Description
R	Ref. Point Id	Identification of a reference point in the case the sensor is mounted on the trailer
R	Sensor Position X offset	Mounting position of the sensor in the x position w.r.t. the reference point of the vehicle
R	Sensor Position Y offset	Mounting position of the sensor in the y position w.r.t. the reference point of the vehicle
R	Sensor Position Z offset	Mounting position of the sensor in the z position w.r.t. the reference point of the vehicle
R	Range	Value of the sensor range
R	Horizontal opening angle start	Start of the horizontal opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Horizontal opening angle end	End of the horizontal opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle start	Start of the vertical opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle end	End of the vertical opening angle of the sensor w.r.t. a vehicle body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
Vehicle Sensor Container		

Importance	Field Name	Description
R	Sensor Position X offset	Mounting position of the sensor in the x position w.r.t. the reference point of the infrastructure
R	Sensor Position Y offset	Mounting position of the sensor in the y position w.r.t. the reference point of the infrastructure
R	Sensor Position Z offset	Mounting position of the sensor in the z position w.r.t. the reference point of the infrastructure
R	Range	Value of the sensor range
R	Horizontal opening angle start	Start of the horizontal opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Horizontal opening angle end	End of the horizontal opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle start	Start of the vertical opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
R	Vertical opening angle end	End of the vertical opening angle of the sensor w.r.t. the infrastructure body frame. The angle is measured from Horizontal opening angle start to Horizontal opening angle end in counter-clockwise direction
Stationary Sensor Container		

Importance	Field Name	Description
R	Circular	Sensor with a circular view. It provides the radius of the sensor field of view and the center point w.r.t. the vehicle body frame
R	Polygon	This can be used to provide a detection polygonal area. This area can be associate to one sensor or considered as the union of several sensors characteristics. In the latter case, the sensor type should be set to “fusion”. The field PolyPoint provides the geometry of this detection area
R	Ellipse	This field can be used to provide a description of an elliptic detection area. The required information is only the geometry of the ellipse and its orientation in the frame of the infrastructure.
R	Rectangle	This field can be used to provide a description of a rectangular detection area. The required information is only the geometry of the rectangle and its orientation in the frame of the infrastructure.
Detecting characteristics of a sensor		

Importance	Field Name	Description
R	Object ID	Identifier of the detected object. This id is unique for every object from the same user. This id should help in identifying different objects. Before labeling with this id, detected objects should be refined via data fusion and tracking procedures in order to have a consistent estimation of objects motion. It is recommended to use the same id for the same objects in subsequent CPMs to facilitate the association.
R	Sensor ID	Id of the sensor that detected the perceived object
R	Time of Measurement	A timestamp that states the exact time at which the measurements from the detected object have been taken. This must not be confused with the message timestamp. It is possible to express this time relatively to the message timestamp. Information for synchronization should be provided.
R	Object Age	Provides the age of the detected object. In order to have this field, several data association procedures need to be taken into account before sending the perceived objects
R	Object Confidence	The confidence associated to an object. This confidence should be computed in a way that it is equal for every user, i.e. every road entity can have the same information from this value. Objects with confidence under a certain threshold value should not be sent.
R	X, Y, Z Distances	Absolute distance from the detected object to the user reference point in the three coordinates x, y, z at the time of measurement. This distance is expressed in the detecting user reference frame. A confidence level of 95% should be provided.
R	X, Y, Z Speed	Relative speed of the detected object from the user reference point in x, y, z directions at the time of measurement. This parameter should be estimated as well as possible in order to track the object. A confidence level of 95% should be provided.
B	X, Y, Z Acceleration	Relative acceleration of the detected object from the user reference point in the x, y, z directions at the time of the measurement. A confidence level of 95% should be provided.
B	Yaw angle	Relative yaw angle of object from the user reference point. This angle is computed w.r.t. the x direction of the detecting user body frame. A confidence level of 95% should be provided.

B	Object Bounding Box	A bounding box representing the detected object. This object can be modeled as a parallelepiped, with the 3 dimensions Length, Width and Height. Some different and more detailed shapes of the road entity as a mesh or a surface estimation can be considered to be included in this field. We also need to associate a level of uncertainty relative to these 3 measures, in order to quantify risks in estimation of the detected user boundaries.
B	Object reference point	The reference point relative to the perceived object. Provided measurements are computed w.r.t. this point.
B	Object dynamic status	Classification of a perceived object towards the capability to move. Three status are possible:Dynamic Has been dynamic Static
R	Classification	Provides the classification of an object in several pre-defined categories.
Perceived Object Container		

7.1.4.7 CPM Reference Position

For vehicles, we consider the reference position (i.e. the origin of the vehicle body frame) as the center of the front side (i.e the width) of the bounding box of the vehicle, according to the CPM standards. However, there exists several models in literature that consider the origin of the vehicle body frame placed on the middle of the rear wheels axis. Other implementations also suggest also putting it on the middle of the back side of the vehicle bounding box. It is mandatory to define a unique standard for this field. If this is not possible, transformations to pass from the alternative vehicle body frame to the standard one must be provided.

If the user is a RSU, the origin of the local frame should be defined as a point of the infrastructure (e.g. the point in which a camera is mounted).

7.1.4.8 Sensor Mounting Specifications

In Figure 7.8, we can see an example of the sensors parameters that can be described in the Sensor Information container.

7.1.4.9 Sensors Field of View Description

It is possible to describe the field of view of a sensor according to its characteristics. It is also possible to fuse several fields of view obtaining a polygon describing a more complex covering zone.

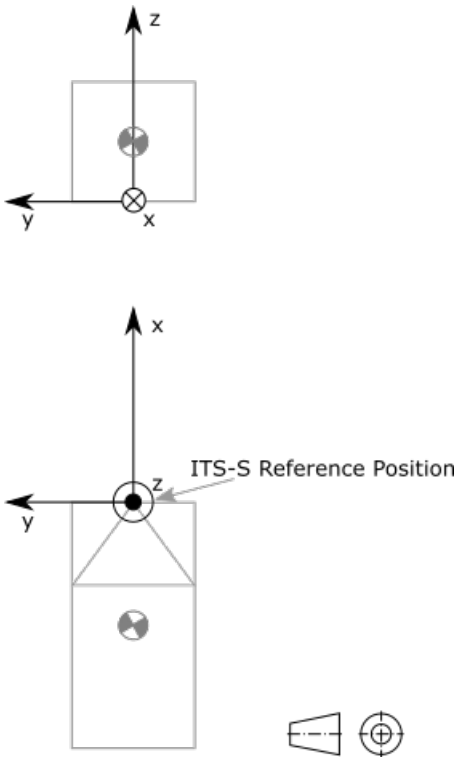
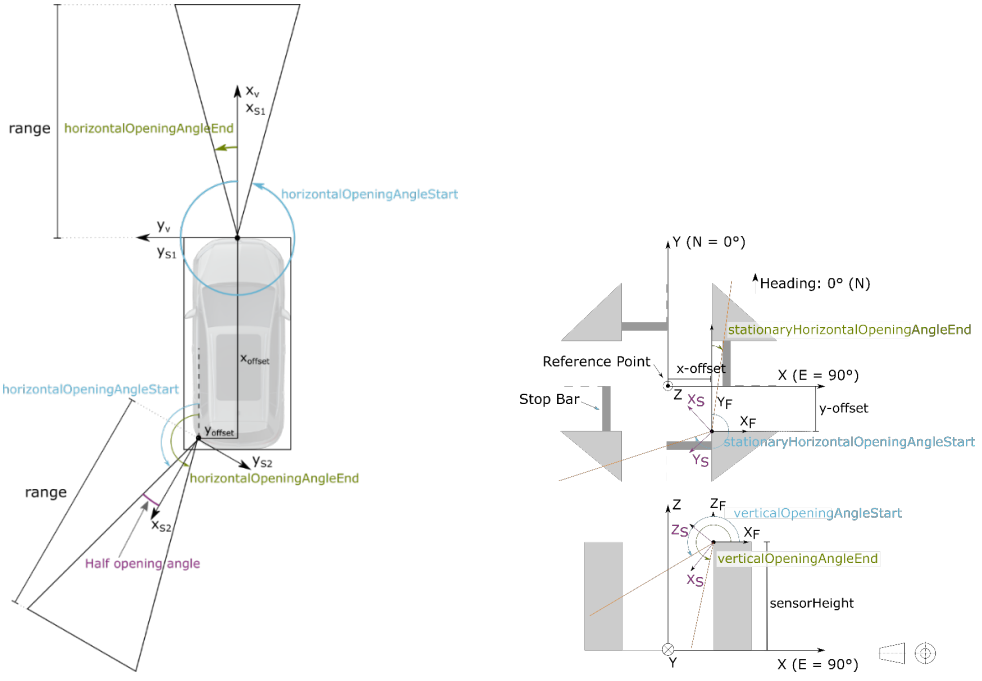


Figure 7.6: Different standards for the body frame.



(a) (b)

Figure 7.7: Sensors mounted on a vehicle and on an intelligent infrastructure with parameters description.

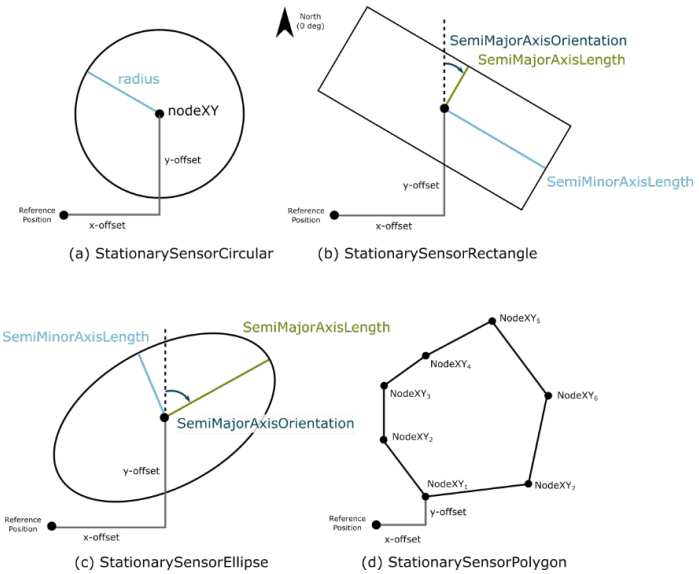


Figure 7.8: Illustration of the different sensors fields of view encoded in CPM.

7.2 Enhanced Perception Datasets for Autonomous Driving

In this part the setup and configuration adopted to record the experimental dataset are described.

7.2.1 System Setup and Sensors

The dataset has been recorded via a fleet of three Zoe, each one equipped with proprioceptive and exteroceptive on board sensors; a truck mounting a precise localization system giving its ground truth position and an infrastructure with a camera providing additional knowledge on road environment.

The localization sensors are:

- NovAtel SPAN-CPT: It provides a centimeter level of precision by the use of a fixed antenna providing RTK corrections. It is mounted on all the fleet of road vehicles.
- Ublox 8T: It provides a precision of the order of a meter. It is mounted on the three Zoe.
- Septentrio Aste-RX SB: It provides a precision of the order of a meter but a better quality in the acquired measurements. It is mounted on the three Zoe.

Obstacles and road agents detection is provided by:

- Velodyne VLP32-C: A LiDAR sensor representing the environment surrounding the vehicle on which it is mounted in terms of a dense point cloud. It is mounted on the three Zoe.

The infrastructure is composed by a monocular camera. The acquired images are processed to identify and classify the road obstacles via a neural network based object detector algorithm.

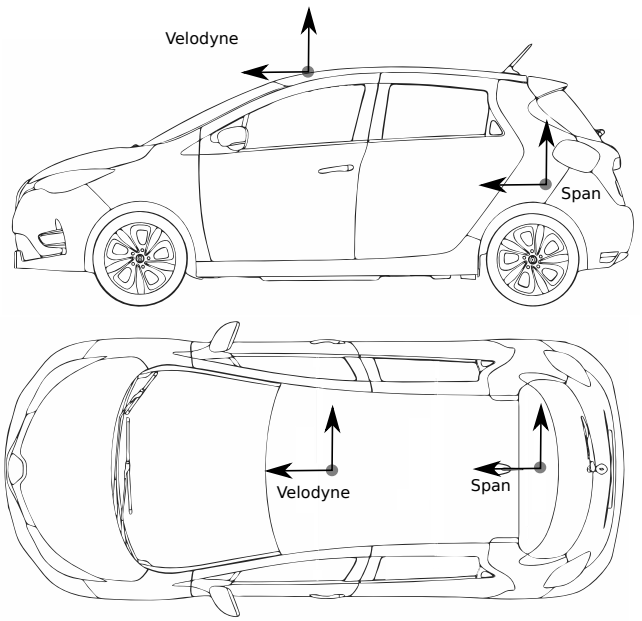


Figure 7.9: Sensors reference frames position

7.2.2 HD Map

In this work a precise driving environment representation is provided. Such a representation is encoded into a high definition (HD) map containing relevant information regarding road geometry and lane level topology. The map results having a centimeter level of precision, which gives a precise information about the position of roads and road elements in every scenario. In particular, road environment is represented by:

- Centerlane_polyline: it is an oriented polyline used to represent the middle of each lane. The driving direction is from the first to the last point of the polyline. Each of them has a unique ID, a list of points which represent the road geometry and the width of the lane.
- Laneborder_polyline: it is a polyline used to represent the left and right borders of each lane. Notice that a road can have several lanes and a couple of laneborder_polyline for each of them. Each object present a unique ID and a set of points representing its geometry.
- Road_element: This object represents all the relevant road entities being present in the driving environment. In particular, objects as pedestrian crossings, parking spots and speed bumpers are encoded as polygons and their corresponding occupancy can be retrieved in the Geometry attribute.
- Road_sign: In this object, the position and type of road signs are encoded. Each of them presents a unique ID, a type and a point representing its position.

The information about each of the objects is contained into a .csv file with the same name of the referenced object and the same properties explained before. Furthermore, to read handle and visualize the contents of each .csv file, the script display_map.py is provided. In this script, contents from each .csv file are read and stored in a dictionary data structure. If the reader is interested in importing map information into a ROS environment, it is necessary to compile in a ROS workspace the package “map_visualization”. After that, it has to launch

Sensor	x	y	z
Velodyne	1.736	0.128	1.296
Span	0.023	-0.005	1.296
Ldmrs	3.377	0.000	0.075
Laser: lms	-0.807	0.000	0.160
GPS ant	1.360	-0.003	1.300
Mobileye	2.060	0.000	0.860
Dashcam	2.220	0.300	0.860

Table 7.3: White Zoe sensors positioning w.r.t. fixed *base* reference frame

Sensor	x	y	z
Velodyne	1.736	0.128	1.296
Span	0.023	-0.005	1.296
Ldmrs	3.377	0.000	0.075
Laser: lms	-0.807	0.000	0.160
GPS ant	1.360	-0.003	1.300
Mobileye	2.060	0.000	0.860
Dashcam	2.220	0.300	0.860

Table 7.4: Grey Zoe sensors positioning w.r.t. fixed *base* reference frame

the "map_visualization" node and subscribe to the topics "/Centerlane_polylines", "/Laneborder_polylines", "/Road_elements" and "/Road_signs" to retrieve information about the corresponding quantities.

7.2.3 Reference Frames

All the sensors reference frames are depicted in Figure 7.9, the sensors disposition is the same for all the fleet. Relatives distances between sensors are represented in Tables 7.3, 7.4, 7.5, one for each vehicle of the fleet. Only the sensor mounted on the specific vehicle are shown. Sensors distances are written w.r.t. *base* reference frame, the x dimension increases along the vehicle, the y dimension increases through its left.

7.2.4 Dataset Organization

7.2.4.1 Architecture

To facilitate the use of the dataset, the general architecture and the main blocks composing it are described in detail. First, the dataset is mainly divided in scenar-

Sensor	x	y	z
Velodyne	1.736	0.128	1.296
Span	0.023	-0.005	1.296
Ldmrs	3.377	0.000	0.075
Laser: lms	-0.807	0.000	0.160
GPS ant	1.360	-0.003	1.300
Mobileye	2.060	0.000	0.860
Dashcam	2.220	0.300	0.860

Table 7.5: Blue Zoe sensors positioning w.r.t. fixed *base* reference frame

ios. Each scenario represents a group of recordings over a fixed time horizon. In particular, all the recordings that have been grouped in the same scenario deal with the same driving situation (e.g. roundabout crossing, platoon navigation etc...). Scenarios have been chosen to capture driving situations or maneuvers that can be improved or optimized exploiting V2V and/or I2V-V2I communication.

For every scenario, several instances containing data recordings about the corresponding driving situation are provided. Each instance is different w.r.t. the others because the distribution of traffic participants and the scenario configuration may change. For instance, several recordings of the same use-case with different traffic density and different weather conditions are given. This configuration allows a user to have a wide set of instances of the same scenario to be used as a test bench for user-implemented algorithms.

7.2.4.2 Data Layers

One of the main advantages of this dataset is that information about data can be found at different abstraction levels. In other words, different representations are provided for a specific group of data. This is particularly useful for sensors data. The raw data can be interesting for users that want to manipulate data at the sensor-level. For users that want to exploit the final output of sensor data (e.g. the bounding boxes of a detection algorithm), they do not have to implement themselves the whole pipeline to obtain, from sensor-level data, the final output. In this dataset, both the raw-data and the final treatment, corresponding to the application of a certain processing method on the raw data, are provided for many sensors. This choice is motivated by the fact that in the case of V2V and V2I-I2V, it is often preferred to exchange data between communicating traffic participants in a compact object-like representation instead of raw data. This concept can be observed in the main definitions of V2V, V2I-I2V messages (e.g. CAM, DENM and CPM).

Finally, it has to be underlined that for the non raw-data layers, the output has been obtained considering some state-of-the-art algorithms that the authors found in the existing literature e.g. YOLO [45]. One consequence is that the treatment output depends on the particular algorithm used to compute it. Moreover, each algorithm may have some weaknesses that are necessarily included in the provided layer. For the sake of completeness, the authors provide references as [45] and [**?**] about the used algorithms in each non raw-data layer.

7.2.4.3 Data Format

The main format that is used as a container for the data is the Rosbag format. This format has several advantages: first, it is compatible with the widely used middleware ROS. This allows to exchange messages simulating real-time communication that mimic the real Vehicle-to-vehicle and Vehicle-to-Infrastructure iteration. Moreover Rosbags can be easily replayed in order to simulate data and their real time acquisition from the sensor or the subsequent reception from the suited algorithm processing them.

7.2.5 Scenarios

An overview and a brief description of the scenarios available in the dataset is provided. Each scenario has been recorded from real driving situations.

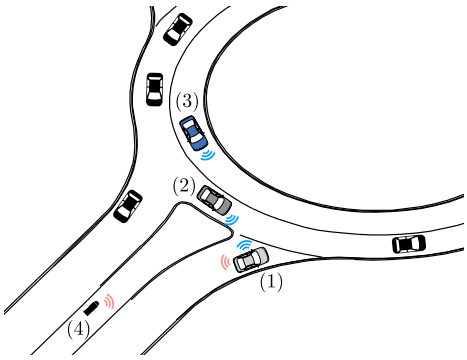


Figure 7.10: A representation of the Roundabout Crossing scenario. The three Zoe are depicted respectively in white (1), grey (2) and blue (3), while other traffic participants are depicted in black. Also the intelligent infrastructure (4) is represented.

7.2.5.1 Roundabout Crossing

The aim of this scenario is to re-create the navigation of an autonomous driving (AD) vehicle through a roundabout. In particular, the focus is in how the V2V-I2V can improve the on-board AD vehicle perception by extending its field of view to achieve the navigation task. In the presented case, the white Zoe has been used as an autonomous vehicle, while the other vehicles (gray and blue ones) as other traffic participants communicating with it. Furthermore, the information from the intelligent infrastructure is obtained from of a remote camera pointing towards the roundabout.

This configuration has several advantages. First, the exact position and speed (with low uncertainty) of every Zoe is precisely known.

Then, not only V2V perception, communicated from other traffic participants, can be exploited, but also information provided by the intelligent infrastructure by I2V communication. Figure 7.10 illustrates this use-case.

7.2.5.2 Navigation with FOV Partially Occulted

In this scenario, the benefits of V2V and I2V communication to extend the field of view of an AD vehicle driving behind a truck are exploited. A vehicle platoon composed by a truck between two Zoe is considered as depicted in Figure 7.11. In such a case, the field of view of the white vehicle is limited by the presence of the truck. However, thanks to V2V communication, the first Zoe can send information about its perceived environment back to the white one. A wide set of this information is complementary w.r.t. the one of the white Zoe because of the absence of any obstacle in the blue Zoe field of view. Furthermore, when the platooning is driving in proximity of the intelligent infrastructure, the knowledge of the AD vehicle is enhanced by the camera perception. This allows the AD vehicle to have a full comprehension of the surrounding environment even if mostly covered by the vehicle in front.

7.2.5.3 Roundabout Insertion with FOV Partially Occulted

This scenario is provided to study the benefits of V2V and I2V communication in the case of a hazardous driving maneuver. As it can be seen in Figure 7.12, the white Zoe and the van are performing a roundabout insertion maneuver at the same time on different parallel lanes of the same road. In this situation, the FOV of the

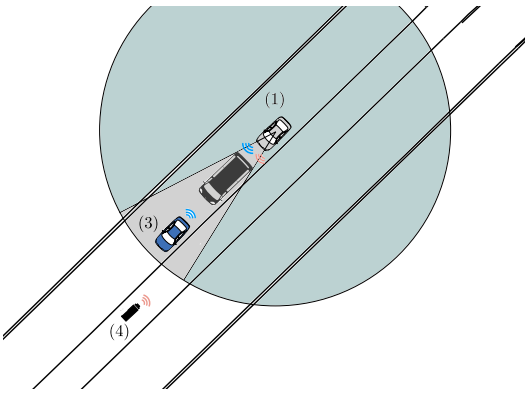


Figure 7.11: A platooning scenario where the view of the vehicle behind the truck (1-white Zoe) is partially reduced (FoV in light blue, occulted part in grey). In such case, V2V information transmitted backward by the blue Zoe (3) along with I2V communication (4-camera) can help the white Zoe to see through the truck.

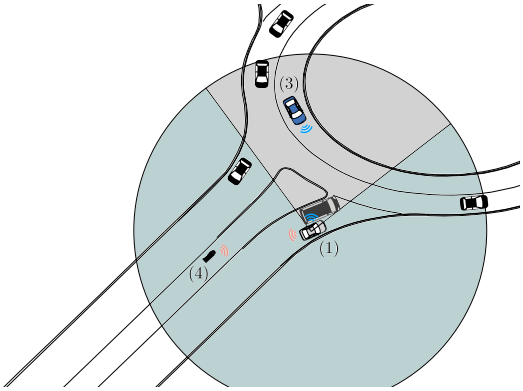


Figure 7.12: An hazardous roundabout insertion maneuver with the field of view in light blue of the white Zoe (1) partially occulted by the presence of a van (occlusion in grey).

white Zoe is partially occluded by the van. Even the AD vehicle cannot directly perceive vehicles in the roundabout it waits at the sign if the insertion cannot be safely performed. On the contrary if the insertion can be executed without risks the AD continues its movement even if its field of view is occluded during the whole maneuver.

The main issue in this maneuver is the lack of information in the roundabout and a better knowledge of the occulted zones can be obtained by exchanging perception with other vehicles having an on-board perception system.

7.2.5.4 Hidden Pedestrian Road Crossing

This scenario is presented in order to show the benefits of shared perception in terms of different points of view of the same scene. As described in Figure 7.13 the field of view of the white Zoe, being the AD vehicle, is partially covered by a van on the right lane of the same road, hiding a pedestrian crossing the street. The information exchange between the communicating vehicles and between the AD vehicle and the smart infrastructure allows the intelligent one to stop before the pedestrian crossing even if it is not capable of directly detecting the pedestrian. It can be seen how collaborative perception, providing different fields of view of the scene, increases situational awareness and allows to have a better comprehension of

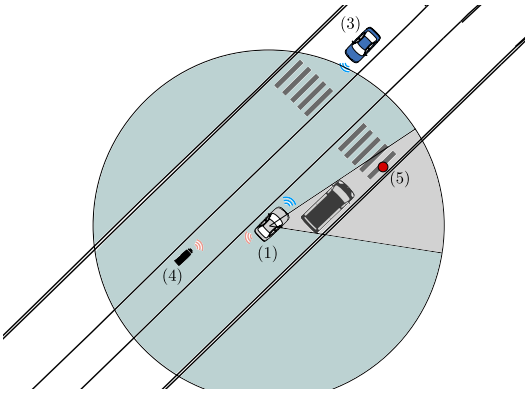


Figure 7.13: A dangerous road crossing of a pedestrian in red (5) with the field of view depicted in light blue of the white Zoe (l1) partially occulted by the presence of a van (occlusion in grey).

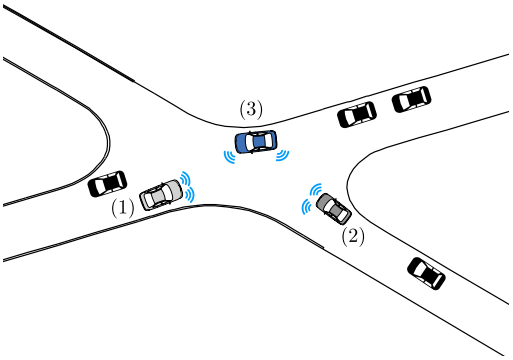
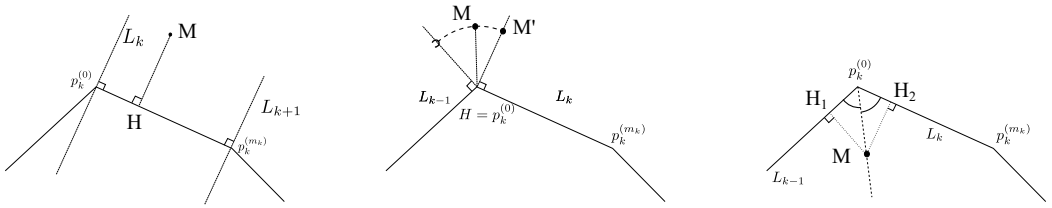


Figure 7.14: An example of intersection crossing exploited in normal navigation. The three Zoe are depicted in white (1), grey (2) and blue (3) while the other vehicles in black. V2V communication is performed among the vehicles of the fleet.



(a) Orthogonal projection for a point M between the segment L_k .

(b) Distance of point M during the transition between L_{k-1} and L_k.

(c) A jump appears for point M when it crosses the bisecting line at point $p_k^{(0)}$.

Figure 7.15: A Polyline map-matching with a point-to-segment distance.

the environment. Consequently, because what cannot be directly seen is perceived anyway, filling the AD vehicle lack of information, the safety of the autonomous navigation is increased.

7.2.5.5 Navigation

This scenario has been realized to provide the best possible scenario not based on a particular situation. In this scenario the whole fleet composed by the three Zoe is used to perform normal navigation. Different scenarios such as roundabouts or intersections are traversed as long as urban or suburban environments and road previewing different maximum speeds. When a vehicle meets the other, the information exchange takes place enhancing their respective fields of views and providing different fields of view of the scene. When a vehicle of the fleet enters in the intelligent infrastructure field of action, the shared perception is performed. A possible example of such navigation and interaction is depicted in Figure 7.14.

7.3 Road Centerlane Representation Models

In this appendix, the different methods presented in Chapter 2 to model the road centerlane are presented, highlighting the technical details and the implementation choice that we used throughout the manuscript.

7.3.1 Polyline Model

The main idea of this approach is to connect two consecutive points $p_k^{(i)}$ and $p_k^{(i+1)}$ with an oriented segment. Such process is repeated for every couple of subsequent points of the map. To compute curvilinear coordinates with this new map representation, the map-matching is performed by searching the smallest distance among all the distances from point M to each segment of the polylines. To compute curvilinear coordinates with this new map representation, the map-matching is performed by searching the smallest distance among all the distances from point M to each segment of the polylines. However, in the case of a complex road networks, it is possible that sometimes there are more than one candidate points for map-matching the point M. In such cases, this basic map-matching procedure has to be robustified using heading information or allowing multiple hypothesis for the matching in case of ambiguity.

The computation of the distance from a point to a segment is illustrated in Figure 7.15. When the point M is between the orthogonal line of the starting point of the segment L_k and the orthogonal line of the end point of L_k, the obtained distance is the orthogonal distance and the map-matching procedure is straightforward. Such case is illustrated in Figure 7.15a.

On the other hand, when the point is in the zone between the orthogonal line of the end point of the previous segment L_{k-1} and the orthogonal line of the starting point, the distance is always computed from point M to point $p_{k-1}^{(m_{k-1})}$. Notice that the point $p_{k-1}^{(m_{k-1})}$ is the same as $p_k^{(0)}$as illustrated in Figure 7.15b. Moreover, one can also observe that point H obtained with the map-matching result is either the orthogonal projection in Figure 7.15a or the starting point p_i in Figure 7.15b, depending on the position of point M.

This point is used to compute the curvilinear abscissa, summing the lengths of each segment from the first one of the road link to the map-matched one as we said in section 2.3.1. The curvilinear ordinate is obtained as the signed lateral distance computed during the map-matching step and ψ the curvilinear orientation is the difference between the orientation in the world frame and the orientation obtained from the map-matched segment.

However, one can see that in the case as the one depicted in Figure 7.15b, the transformation from Cartesian coordinates to curvilinear coordinates and vice versa is not bijective.

This comes from the fact that situations where discontinuities and stationary points may arise because, in general, polylines are not differentiable at their junctions points. This implies that the discontinuity of the tangent in such points leads to discontinuity issues near these conjoining points. If we observe carefully the case shown in Figure 7.15b, we can see that the value of the curvilinear abscissa remains the same for every point M that lies in the zone between the lines L_{k-1} and L_k. This happens because if we use map-matching on this points, the projection always corresponds to the projected point M'. On the other hand, Figure 7.15c depicts a case where two projections can appear at the same time during map-matching. This

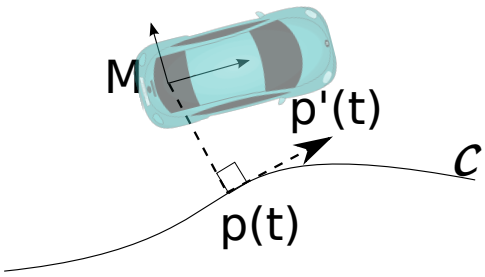


Figure 7.16: Map-matching process for a point M w.r.t. a spline $\mathcal{C}$ defined by the vector of polynomial $p(t)$.

depends on where point M lies with respect to the bisecting line. In other words, we can observe that, once the bisecting line is crossed, a jump appears in the curvilinear abscissa value, i.e. a discontinuity appears. The same behavior can be also seen in the case of the curvilinear orientation, because such quantity is computed w.r.t. to the segment.

7.3.2 Spline Model

One way to remove the discontinuity problem discussed in the previous part is to use a different curve representation instead of a polyline-based one. In particular, we consider a polynomial representation obtained by fitting the set of points $\left\{p_k^{(i)} = [x_i,\, y_i]^T\right\}$. The goal is to obtain a smoother curve than polylines and to eliminate discontinuities at polyline vertex. To do so, each segment is replaced by vectors of two polynomial functions representing a curve to obtain at least a C^1 continuity. In this work, we consider only polynomials up to the 3^{rd} degree, which correspond to cubic splines.

The spline curve can be defined as follows:

$$\left\{p_i(t) = \begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix}, i \in \{0, ..., n\}, t \in [0; 1]\right\}, \tag{7.1}$$

where $p_{i,x}(t)$ and $p_{i,y}(t)$ represent two polynomials curves parameterized by t. Every point of the curve corresponds to a value of the vector $p_i(t)$. Considering this representation, and exploiting the continuity of the curve, the tangent at point $p_i(t)$ is equal to the first derivative $p_i'(t)$, while the curvature is equal to the second order derivative $p_i''(t)$.

In order to find the closest point of the spline from point $M = [x,\, y]^T$, a polynomial equation needs to be solved. In particular, all the minimum and maximum points can be found when the condition $(M - p_i(t)) \perp p_i'(t)$ is met. In other words, this leads to the resolution of the following equation:

$$(M - p_i(t)) \,.p_i'(t) = 0. \tag{7.2}$$

However, this equation has no closed form solution in the general case. To obtain the result faster, we first start by searching the closest vector of polynomials corresponding to an index i_{min}. In such a way, we only have to find the roots for this single index. To find this index i_{min}, the distance between M and every joint point $p_i = p_i(0)$ has to be computed. When the minimal distance is found for an index i, the index is $i_{min} = i$ if the projection of M in the tangent at the joint point $p_i' = p_i'(0)$ is positive and $i_{min} = i - 1$ otherwise.

To better achieve this task, the dot product is performed between polynomial of 3^{rd} degree vector $M - p_i(t)$, and the polynomial of 2^{nd} degree vector $p'_i(t)$. This leads to a vector of polynomials of degree 5. As we previously said, there exists no general closed-form solution to find the roots of a polynomial of degree 5. However, numerical approaches exist to solve this problem. In our problem, we do not consider neither complex roots nor the solutions not contained in the [0, 1] interval. Finally, we select at the end the parameter $t_{\min}$ in a way that the quantity $\|M - p_{i_{\min}}(t_{\min})\|$ is the smallest distance between the point M and the curve $\mathcal{C}$. This distance will also be considered as the curvilinear ordinate n. A minus sign is added if the point M is on the right side of the tangent $p'_{i_{\min}}(t_{\min})$.

To compute the curvilinear abscissa length s, we need to integrate the quantity $\left\|p'_{i_{\min}}(t)\right\|$ according to the following formula:

$$s = \int_0^{t_{\min}} \left\|p'_{i_{\min}}(t)\right\| dt + \sum_{j=0}^{i_{\min}-1} \left(\int_0^1 \left\|p'_j(t)\right\| dt \right). \tag{7.3}$$

Equation 7.3 has no closed-form solution, because computing the norm of a 2^{nd} degree polynomial leads to compute the square root of a 4^{th} degree polynomial. As we previously pointed out, only an approximated solution can be obtained in this case. In order to make the computation faster, one can compute in advance the arc length associated to each p_j. Doing this, only the first term of Equation (7.3) needs to be computed on-the-fly.

On the other hand, if one wants to find the Cartesian pose that corresponds to a given curvilinear pose, the inverse of Equation (7.3) needs to be solved. This means that one needs to compute the correct value of $t_{\min}$ given the curvilinear abscissa s. Again, numerical approaches need to be used to solve this problem. An alternative solution could be to approximate the inverse and the arc-length by a polynomial or another function [94].

In order to compare different possible behaviors, two different cubic splines are studied in this section. The first is the B-Spline, which consists in an approximation of the map which gives a smooth trajectory to follow. The latter is Hermite spline, which is constructed with an interpolation of the map. For this reason, it provides a better representation of the map. However, this last spline model is nevertheless less comfortable if the control points are not well positioned.

7.3.2.1 B-Spline Model

A B-Spline is an approximation of the path which is not passing through the points of the path $\{p_i,\, i \in \{1, ..., n\}\}$. One can compute the polynomials of a B-Spline as follows [15]:

$$\begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix} = \begin{pmatrix} p_{i-1} & p_i & p_{i+1} & p_{i+2} \end{pmatrix} \cdots \times \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}. \tag{7.4}$$

The advantages of using an approximation is to have a smoother curve more comfortable for control. However, this approximation can be problematic in some cases, e.g., the spline may cut through the corner at T-junctions.

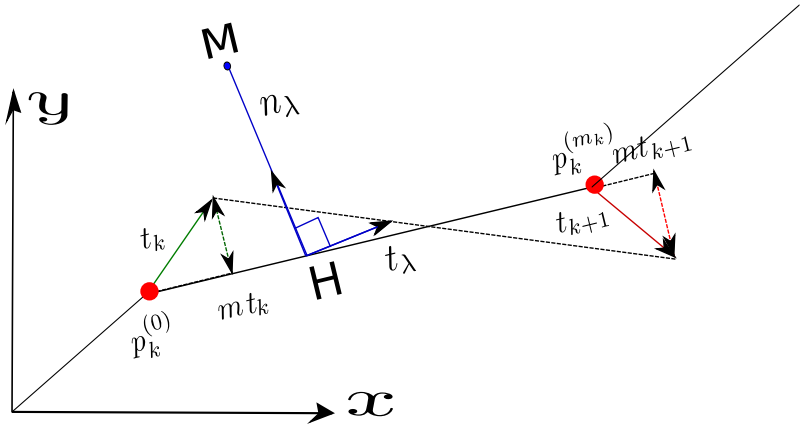


Figure 7.17: Lanelet map-matching of point M on a polyline segment.

7.3.2.2 Hermite Spline Model

In the case where one needs a curve going through all the points of the polyline, an interpolation can be used instead. This can be done using a Hermite spline with polynomials defined as follows [54]:

$$\begin{pmatrix} p_{i,x}(t) \\ p_{i,y}(t) \end{pmatrix} = \begin{pmatrix} p_i & t_i & p_{i+1} & t_{i+1} \end{pmatrix} \cdots \times \begin{pmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}, \tag{7.5}$$

where $\{t_i,\, i \in \{1, ..., n\}\}$ are the tangents of these control points. They can be computed as $t_i = (p_{i+1} - p_{i-1})/2$.

7.3.3 Lanelet Model

Another strategy to overcome the problem of discontinuities in the neighborhood of the junction points is to use a map-matching algorithm based on a non-Euclidean metric as shown in Figure 7.17. This leads to the computation of a new frame, called lanelet frame [11], where the matched point on the polyline segment is obtained as a convex combination of the starting and ending points of the given segment, and not with a straightforward point-to-line distance, as in the previous case. The main idea of this technique is to define the normal and tangent vectors to a discrete curve, in our case a polyline-based one. In this case, such vectors are identified as n_λ and t_λ, and they constitute the abscissa and ordinate axis of the lanelet frame. The matching point w.r.t. a given link segment can be computed as follows:

$$p_\lambda = \lambda p_k^{(i)} + (1 - \lambda) p_k^{(i+1)}. \tag{7.6}$$

where $0 \leq \lambda \leq 1$. In order to obtain an orthogonal reference frame, we add the constraint that n_λ and t_λ, that are respectively the tangent and normal vectors to the curve in p_λ (i.e. the abscissa and the ordinate axis in the new frame), must be orthogonal, i.e. $n_\lambda \cdot t_\lambda = 0$. From this constraint, one can compute λ as follows:

$$\lambda = \frac{x + y \cdot mt_k}{\ell - y(mt_k - mt_{k+1})}, \tag{7.7}$$

where mt_k are the ordinates of the tangent vectors at points $p_k^{(i)}$ in the form $t_k = (1, mt_k)$ and $t_{k+1} = (1, mt_{k+1})$ computed in the local frame of the k_{th} segment of the link and ℓ is the length of the polyline segment. Once the origin of the frame p_λ has been computed, we compute the new frame abscissa axis called t_λ again as a convex combination of two vectors called respectively t_k and t_{k+1} which have been previously computed for every polyline segment:

$$t_\lambda = \lambda t_k + (1 - \lambda) t_{k+1}. \tag{7.8}$$

Finally, we compute the normal vector n_λ, orthogonal to t_λ, as $n_\lambda = M - p_\lambda$. After that, one can obtain the distance from point M to the k_{th} polyline segment considering the value of $||n_\lambda||$ as a thresholding criterion for choosing the polyline segment that better fits the matching.

It can be shown that with this method there are no discontinuities at the polylines junction points. As for the polyline method, this technique produces also a partition of the space, i.e., a set of regions in which all the points are matched to a certain polyline segment. Such region are determined by computing the normal vector for every tangent vector in every junction point of the structure and using them as a boundaries.

However, the aforementioned statement holds under the constraint that the point M that needs to be matched is close enough to the polyline segments. Otherwise, it might happen that some segments will be skipped during the matching step, leading again to discontinuities in the curvilinear abscissa.

The conversion from Cartesian coordinates to curvilinear ones and vice versa is straightforward in this context. Once the index k of the matched polyline segment is known, we compute the curvilinear abscissa as follows:

$$s = \sum_{j=1}^{k-1} \ell_j + \lambda \ell_k \tag{7.9}$$

where ℓ_j is the length of the j_{th} polyline segment. Regarding the value of n, we compute it by considering the second component of the vector n_λ in the lanelet frame. In fact, this quantity corresponds to the lenght of the ordinate axis of such a frame.

On the other hand, if one want to convert a curvilinear pose $[s, n, \psi]$ back to the Cartesian coordinates, the corresponding λ is computed as follows:

$$\lambda = \frac{s - \sum_{j=1}^{k-1} \ell_j}{\ell_k}, \tag{7.10}$$

where the index k that indicates the k^{th} segment is found by subtracting iteratively from s the length of the j^{th} segment of the polyline, until it becomes negative or null.

Given λ and the index k of the matching polyline, it is easy to compute again H_λ and t_λ and, with the knowledge of t_λ, it is possible now to compute the coordinates of the matched point in the local frame as $M_l = R \cdot t_\lambda / ||t_\lambda|| \cdot n$, where $R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the rotation matrix computed for a rotation angle of $\frac{\pi}{2}$. After this computation, the point M_l is transformed in the world frame. It is important to underline that once the criterion for matching the polylines is fixed, this method is bijective and easy to compute.

Bibliography

[1] Intelligent transport systems (its) vehicular communications basic set of applications part 2: Specification of cooperative awareness basic service. 09 2014. 157, 159, 166

[2] V2x communications message set dictionary j2735. 07 2020. 158

[3] Asvadi A., Peixoto P., and Nunes U. Detection and tracking of moving objects using 2.5d motion grids. *IEEE 18th International Conference on Intelligent Transportation Systems*, pages 788–793, 2015. 77, 78

[4] J. Al Hage, P. Bonnifait, P. Xu, and J. Ibanez-Guzman. Localization integrity for intelligent vehicles through fault detection and position error characterization. *IEEE Transactions on Intelligent Transportation Systems*, 10 2020. 15

[5] J. Alonso, V. Milanés, J. Pérez, E. Onieva, C. González, and T. de Pedro. Autonomous vehicle control systems for safe crossroads. *Transportation Research Part C*, pages 1095–1110, 2011. 21

[6] M. Althoff, M. Koschi, and S. Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *IEEE Intelligent Vehicles Symposium*, pages 719 – 726, 2017. 22

[7] S. Aoki, T. Higuchi, and Onur A. Cooperative perception with deep reinforcement learning for connected vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 328–334, 2020. 100

[8] B. Qadeer Baig, M. Perrollaz, and C. Laugier. Advances in the bayesian occupancy filter framework using robust motion detection technique for dynamic environment monitoring. *IEEE Robotics and Automation Magazine*, 03 2014. 22

[9] L. Banjanovic-Mehmedovic, E. Halilovic, I. Bosankic, M. Kantardzic, and S. Kasapovic. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory. *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 8, 2016. 21

[10] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: exact solution. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):769–777, 2002. 125

[11] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation for autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 420–425, 06 2014. 24, 185

[12] E. Bernardi. Lidar based vehicle detection with high-definition maps. 02-08 2019. 74

[13] E. Bernardi, S. Masi, P. Xu, and P. Bonnifait. High integrity efficient lane-level occupancy estimation of road obstacles through lidar and hd map data fusion. In *IEEE Intelligent Vehicles Symposium*, page in proceedings, 06 2020. 18, 57, 69

[14] P. Bonnifait, J. Laneurit, C. Fouque, and G. Dherbomez. Multi-hypothesis Map-Matching using Particle Filtering. In *16th World Congress for ITS Systems and Services*, pages 1–8, Stockholm, Sweden, September 2009. 8 pages. 106

[15] C. Boor. *A Practical Guide to Splines*. Springer New York, 12 2001. 184

[16] A. Börcs, B. Nagy, and C. Benedek. *Dynamic 3D Environment Perception and Reconstruction Using a Mobile Rotating Multi-beam Lidar Scanner*, pages 153–180. Springer International Publishing, 2015. 79

[17] A. Börcs, B. Nagy, and C. Benedek. Minimum-area rectangle containing a set of points. 2016. 79

[18] E. Capellier, V. Cherfaoui, and D. Franck. Application of machine learning techniques for evidential 3d perception, in the context of autonomous driving. 01 2020. 15

[19] D. Carlino, S. D. Boyles, and P. Stone. Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems*, pages 529–534, 10 2013. 21

[20] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–17, 03 2015. 21

[21] L. Chen, Q. Li, Q. Zhu, M. Li, and A. Nuchter. 3d lidar point cloud based intersection recognition for autonomous driving. In *IEEE Intelligent Vehicles Symposium*, pages 456–461, 2012. 76

[22] P Chu, Seoungjae Cho, S Sim, K Kwak, and Kyungeun Cho. A fast ground segmentation method for 3d point cloud. *Journal of Information Processing Systems*, pages 491–499, 2017. 77

[23] A. Colombo and D. Del Vecchio. Efficient algorithms for collision avoidance at intersections. *In Proceedings of the 15th ACM HSCC, pp. 145–154, New York, USA, 2012. ACM.*, 2012. 21

[24] Wikibooks contributors. Algorithm implementation/geometry/convex hull/monotone chain. *Wikibooks, The Free Textbook Project*. 82

[25] S. Das and S. Sarvottamananda. Computing the minkowski sum of convex polytopes, 2018. 88

[26] A. de La Fortelle and X. Qian. Autonomous driving at intersections: combining theoretical analysis with practical considerations. In *ITS World Congress*, 10 2015. 21

[27] E. Debada, L. Makarem, and D. Gillet. A virtual vehicle based coordination framework for autonomous vehicles in heterogeneous scenarios. In *IEEE International Conference on Vehicular Electronics and Safety. Vienna, Austria*, 06 2017. 21

[28] V. Desaraju, H. C. Ro, M. Yang, E. Tay, S. Roth, and D. Del Vecchio. Partial order techniques for vehicle collision avoidance: Application to an autonomous roundabout test-bed. *IEEE International Conference on Robotics and Automation*, pages 82–87, 05 2009. 21

[29] Moosmann F. and Stiller C. Joint self-localization and tracking of generic objects in 3d range data. *IEEE International Conference on Robotics and Automation*, pages 1146–1152, 2013. 101

[30] P. Falcone, M. Ali, and J. Sjoberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12:1352–1361, 12 2011. 22

[31] F. Fayad and V. Cherfaoui. Tracking objects using a laser scanner in driving situation based on modeling target shape. In *IEEE Intelligent Vehicles Symposium*, pages 44 – 49, 07 2007. 101, 106

[32] M. Figueiredo, M. Rossetti, R. Braga, and L. Reis. An approach to simulate autonomous vehicles in urban traffic scenarios. In *12th International IEEE Conference on Intelligent Transportation Systems*, pages 1 – 6, 11 2009. 30

[33] T. Fleck, S. Ochs, M. R. Zofka, and J. M. Zollner. Robust tracking of reference trajectories for autonomous driving in intelligent roadside infrastructure. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1337–1342, 2020. 100, 101

[34] D. Frossard and R. Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642, 2018. 75

[35] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao. Are we hungry for 3d lidar data for semantic segmentation? a survey and experimental study. 2020. 75

[36] M. Garzon and A. Spalanzani. An hybrid simulation tool for autonomous cars in very high traffic scenarios. In *15th International Conference on Control Automation Robotics and Vision*, pages 803–808, 11 2018. 63

[37] J. Ge, S. Avedisov, C. He, W. Qin, M. Sadeghpour, and G. Orosz. Experimental validation of connected automated vehicle design among human-driven vehicles. *Transportation Research Part C: Emerging Technologies*, 91, 06 2018. 22

[38] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann. Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In *IEEE Intelligent Vehicles Symposium*, pages 669 – 676, 06 2009. 22

[39] Zhu H., Meng F., Cai J., , and S. Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, 2016. 75

[40] E. Hery, S. Masi, P. Xu, and P. Bonnifait. Map-based curvilinear coordinates for autonomous vehicles. In *IEEE 20th International Conference on Intelligent Transportation Systems*, 2017. 17, 29, 43, 47

[41] Yeping Hu, Wei Zhan, and Masayoshi Tomizuka. Probabilistic prediction of vehicle semantic intention and motion. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 307–313, 2018. 21

[42] Bogoslavskyi I. and Stachniss C. Fast range image-based segmentation of sparse 3d laser scans for online operation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 163–169, 2016. 78

[43] G. Ibanez, T. Meuser, M. A. Lopez-Carmona, and D. Lopez-Pajares. Synchronous roundabouts with rotating prioritysectors (syrops) : High capacity and safety forconventional and autonomous vehicles. page preprints, 2020. 22

[44] Cui J., Liu Y., Xu Y., Zhao H., and Zhao H. Tracking generic human motion via fusion of low- and high-dimensional approaches. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):996–1002, 2013. 102

[45] Farhadi J., Redmon ans A. Yolov3: An incremental improvement. *CoRR*, abs / 1804.02767, 2018. 38, 119, 177

[46] S. Jouannin, L. Trassoudaine, and J. Gallice. Comparison of data association methods. application to road obstacle tracking using a doppler effect radar. *IFAC Proceedings Volumes*, 31(3):489 – 494, 1998. 104

[47] Klasing K., Wollherr D., and Buss M. A clustering method for efficient segmentation of 3d laser data. *IEEE International Conference on Robotics and Automation*, pages 4043–4048, 2008. 78

[48] M. Klischat, O. Dragoi, M. Eissa, , and M. Althoff. Coupling sumo with a motion planning framework for automated vehicles. In *SUMO User Conference*, pages 1–9, 2019. 24, 63, 66

[49] Z. Kokkinogenis, M. Teixeira, P. M. d Orey, , and R. J. F. Rossetti. Tactical level decision-making for platoons of autonomous vehicles using auction mechanisms. In *IEEE Intelligent Vehicles Symposium Paris France. 06 9-12*, 2019. 30

[50] D. Korchev, S. Cheng, Y. Owechko, and K. Kim. On real-time lidar data segmentation and classification. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2013. 76

[51] D. Krajzewicz, G. Hertkorn, C. Feld, and P. Wagner. Sumo (simulation of urban mobility); an open-source traffic simulation. In *4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pages 183–187, 01 2002. 30, 63

[52] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955. 104

[53] He L., Chao Y., and Suzuki K. A run-based two-scan labeling algorithm. *IEEE Transactions on Image Processing*, vol. 17, no. 5:749–756, 2008. 78, 81

[54] C. C. Lalescu. Two hierarchies of spline interpolations. Practical algorithms for multivariate higher order splines. *arXiv:0905.3564 [cs]*, May 2009. arXiv: 0905.3564. 185

[55] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. Christian Gerdes, and M. Pavone. On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions. In *International Symposium on Experimental Robotics*, pages 561–574, 2018. 22, 65

[56] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *10.15607/RSS.2016.XII.042.*, 2016. 75

[57] L. Li and F. Wang. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular Technology*, pages 1712–1724, 11 2006. 21

[58] L. Li and F. Y. Wang. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular Technology, vol. 55, no. 6, pp.1712-1724*, 2006. 21

[59] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 119

[60] Z. Luo, S. Habibi, and M. Mohrenschildt. Lidar based real time multiple vehicle detection and tracking. *International Journal of Computer and Information Engineering*, pages 1125–1132, 2016. 102

[61] Chang M., Lambert J., Sangkloy P., Singh J., Bak S., Hartnett A., Wang D., Carr P., Lucey S., Ramanan D., and Hays J. Argoverse: 3d tracking and forecasting with rich maps. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8740–8749, 2019. 101

[62] Gabb M., Digel H., Muller T., and Henn R. Infrastructure-supported perception and track-level fusion using edge computing. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1739–1745, 2019. 100

[63] Himmelsbach M., Hundelshausen F. v., and Wuensche H. Fast segmentation of 3d point clouds for ground vehicles. *IEEE Intelligent Vehicles Symposium*, pages 560–565, 2010. 77

[64] S. Magdici and M. Althoff. Adaptive cruise control with safety guarantees for autonomous vehicles. In *IFAC-PapersOnLine*, volume 50, pages 5774–5781, 07 2017. 21

[65] S. Masi, S. Ieng, P. Xu, and P. Bonnifait. Augmented perception with cooperative roadside vision systems for autonomous driving in complex scenarios. In *IEEE International Intelligent Transportation Systems Conference*, pages 1140–1146, 9 2021. 146

[66] S. Masi, P. Xu, and P. Bonnifait. Adapting the virtual platooning concept to roundabout crossing. In *IEEE Intelligent Vehicles Symposium*, pages 1366–1372, 06 2018. 18, 43, 51, 99

[67] S. Masi, P. Xu, and P. Bonnifait. A curvilinear decision method for two-lane roundabout crossing and its validation under realistic traffic flow. In *IEEE Intelligent Vehicles Symposium*, page in proceedings, 06 2020. 18, 33, 65

[68] S. Masi, P. Xu, and P. Bonnifait. Roundabout crossing with interval occupancy and virtual instances of road users. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, pages 1–13, 2020. 18, 59, 138

[69] A. I. M. Medina, N. V. D. Wouw, and H. Nijmeijer. Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles. In *IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1696–1701, 09 2015. 21, 43, 59

[70] J. Mei, B. Gao, D. Xu, W. Yao, X. Zhao, and H. Zhao. Semantic segmentation of 3d lidar data in dynamic scene using semi-supervised learning. *ArXiv, abs/1809.00426.*, 2018. 75

[71] J. Mei and H. Zhao. Scene context based semantic segmentation for 3d lidar data in dynamic scene. 2020. 75

[72] E. Rauh Muller, R. Castelan Carlson, and W. Kraus Junior. Intersection control for automated vehicles with milp. In *14th IFAC Symposium on Control in Transportation Systems CTS*, volume 49, pages 37 – 42, 2016. 21

[73] M. Naumann, M. Lauer, and C. Stiller. Generating comfortable safe and comprehensible trajectories for automated vehicles in mixed traffic. In *21st International Conference on Intelligent Transportation Systems*, 2018. 22

[74] M. Naumann and C. Stiller. Towards cooperative motion planning for automated vehicles in mixed traffic. *CoRR*, abs/1708.06962, 2017. 22

[75] T. Nguyen and T. Au. A constant-time algorithm for checking reachability of arrival times and arrival velocities of autonomous vehicles. In *IEEE Intelligent Vehicles Symposium Paris France. 06 9-12*, 2019. 21

[76] F. Poggenhans, J. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *21st International Conference on Intelligent Transportation Systems*, 2018. 23, 31

[77] Gan Q. and Harris C. J. Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):273–279, 2001. 101

[78] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In *European Control Conference*, pages 3452–3458, 07 2015. 21

[79] S. Qu, G. Chen, C. Ye, F. Lu, F. Wang, Z. Xu, and Y. Ge. An efficient l-shape fitting method for vehicle pose detection with 2d lidar. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1159–1164, 2018. 79

[80] G. R. De Campos, P. Falcone, and J. Sjoberg. Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing. In *IEEE Conference on Intelligent Transportation Systems*, pages 1456–1461, 10 2013. 21

[81] J. Pérez Rastelli, V. Milanés, T. De Pedro, and L. Vlacic. Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts. *IFAC Proceedings Volumes, pp.13795-13800*, 2012. 21

[82] J. Pérez Rastelli and M. Santos Peñas. Fuzzy logic steering control of autonomous vehicles inside roundabouts. *Applird Soft Computing*, 2015. 21

[83] V. References Vaquero, I. del Pino, Francesc Moreno N., J. Sola, A. Sanfeliu, and Juan. Andrade C. Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–7, 2017. 75

[84] A. Retting Richard, N. Persaud Bhagwant, E. Garder Per, and D. Lord. Crash and injury reduction following installation of roundabouts in the united states. *American Journal of Public Health*, 2001. 21

[85] R. Schubert, E. Richter, and G. Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *International Conference on Information Fusion*, pages 1 – 6, 01 2008. 109

[86] M. Shan, K. Narula, Y. F. Wong, S. Worrall, M. Khan, P. Alexander, and E. Nebot. Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations, 2021. 100

[87] X. Song, J. Cui, H. Zhao, H. Zha, and R. Shibasaki. Laser-based tracking of multiple interacting pedestrians via on-line learning. *Neurocomputing*, 115:92–105, 2013. 102

[88] G. Thandavarayan, R. Alms, J. Gozalvez, M. Rondinone, R. Blokpoel, and L. Lucken. Infrastructure support for cooperative maneuvers in connected and automated driving. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 20–25, 08 2019. 100

[89] A. Tran, M. Kawaguchi, H. Okuda, and T. Suzuki. A model predictive control-based lane merging strategy for autonomous vehicles. In *IEEE Intelligent Vehicles Symposium Paris France.*, pages 594–599, 06 2019. 21

[90] Duy Tran, Weihua Sheng, Li Liu, and Meiqin Liu. A hidden markov model based driver intention prediction system. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 115–120, 2015. 21

[91] V. Vaquero, A. Sanfeliu, and F. Moreno-Noguer. Deep lidar cnn to understand the dynamics of moving vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4504–4509, 2018. 75

[92] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H. Wuensche. Driving with tentacles: Integral structure for sensing and motion. *Journal of Field Robotics*, pages 25–9, 640–673, 06 2008. 22

[93] Zhan W., Sun L., Wang D., Jin Y., and M. Tomizuka. Constructing a highly interactive vehicle motion dataset. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1366–1372, 06 2019. 31

[94] H. Wang, J. Kearney, and K. Atkinson. Arc-length parameterized spline curves for real-time simulation. In *Proc. 5th International Conference on Curves and Surfaces*, pages 387–396, 2002. 184

[95] M. Worner, F. Schuster, F. Dolitzscher, C.G. Keller, M. Haueis, and K. Dietmayer. Integrity for autonomous driving: A survey. *Proceedings of IEEE/ION PLANS*, 43(4):pp. 666–671, 2016. 15

[96] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. 2017. 75

[97] Zhang X., C. Dong W. Xu, and J. M. Dolan. Efficient l-shape fitting for vehicle detection using laser scanners. *IEEE Intelligent Vehicles Symposium (IV)*, pages 54–59, 2017. 78

[98] I. Xausa, R. Baier, O. Bokanowski, and M. Gerdts. Computation of avoidance regions for driver assistance systems by using a hamilton jacobi approach. *Optimal Control Applications and Methods*, 01 2019. 22

[99] P. Xu, G. Dherbomez, E. Hery, A. Abidli, and P. Bonnifait. System architecture of a driverless electric car in the grand cooperative driving challenge. *IEEE Intelligent Transportation Systems Magazine (ITS Mag.)*, pages Vol. 10, Issue 1, pages 47–59, 03 2018. 67

[100] Maalej Y., Sorour S., Abdel-Rahim A., and Guizani M. Tracking 3d lidar point clouds using extended kalman filters in kitti driving sequences. *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018. 102

[101] D. Zermas, I. Izzat, and N. Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In *IEEE International Conference on Robotics and Automation*, pages 5067–5073, 2017. 74, 77, 78, 79, 80, 81

[102] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps, 2019. 22, 31

[103] Y. P. Zhang, Z. R. Deng, and R. Q. Zhang. An efficient approach of convex hull triangulation based on monotonic chain. In *Information Technology Applications in Industry*, volume 263, pages 1605–1608, 2 2013. 82

[104] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *10.1109/CVPR.2018.00472.*, pages 4490–4499, 2018. 75

[105] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F.Stein, F. Erbs, M. Enzweiler, C. Knoeppel, and

E. Zeeb. Making bertha drive an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6:8–20, 10 2015. 22, 23

[106] A. Zyner, S. Worrall, and E. Nebot. Acfr five roundabouts dataset: Naturalistic driving at unsignalised intersections. *IEEE Intelligent Transportation Systems Magazine*, PP:1–1, 09 2019. 22