UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

COOPERATIVE DRIVING FOR COLLISION AVOIDANCE BASED ON COGNITIVE AGENTS

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN SCIENCES ET TECHNOLOGIES DE L’INFORMATION

PAR

GIANCARLO COLMENARES

AVRIL 2018

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

COOPERATIVE DRIVING FOR COLLISION AVOIDANCE BASED ON COGNITIVE AGENTS

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

PAR

GIANCARLO COLMENARES

Membres du jury:

Président	Ahmed Lakhssassi, PhD. UQO.
Directrice de recherche	Ilham Benyahia, PhD. UQO.
Co-directeur de recherche	Marek Zaremba, PhD. UQO.
Examinateurs externes	Brahim Chaib-Draa, PhD. Université Laval.
	Babak Esfandiari, Université Carleton
Examinateur interne	Shamsodin Taheri, PhD. UQO.

AVRIL 2018

À Franshell, Luna et Jesús

REMERCIEMENTS

A Dios Todopoderoso. A Nohelia, la mujer de mi vida, mi fuente de energía, te amo. A Carlo, Chely, Sheylla y Karla, los amo.

Il y a un grand nombre de personnes et institutions qui m’ont soutenu et qui ont fait possible la terminaison de mes études doctorales : Mme. Ilham Benyahia, ma directrice de recherche, son appui inconditionnel, contributions et guidance ont été fondamentaux pour mon apprentissage et la qualité du travail. Au Département d’Informatique et d’Ingénierie, ses professeurs et personnel d’appui, spécialement au Professeur Marek Zaremba. Au Centre de Recherche en Photonique de l’UQO. Au personnel du Bureau du Registraire, du Bureau des étudiants et des échanges internationaux (BÉÉI), des Finances, des Résidences et d’Accueil qui ont été toujours collaborateurs avec mes demandes et ont beaucoup facilité ma vie au Canada.

À *l’Universidad Nacional Experimental del Táchira*, mon *Alma mater*, dont appui au début de ma carrière a été primaire.

À mes amis de vie, mes compagnons, mes collègues, mes frères, vous êtes toujours là : Marcel, Juan, María y José.

Merci à tous.

COOPERATIVE DRIVING FOR COLLISION AVOIDANCE BASED ON COGNITIVE AGENTS

ABSTRACT

By: Giancarlo Colmenares

Equipped vehicles can perceive their environment via on-board sensors; moreover, those with communication capabilities incorporated have the possibility to share that information with their neighbors. These latter, known as connected vehicles, are the basis for cooperative driving, a topic of interest in the Intelligent Transportation Systems domain; cooperative driving uses vehicular communication technologies to transmit and receive interest information, such as: traffic, safety, routing. This investigation is focused in road safety, specifically we are interested in the reduction of collision risk in autonomous vehicles. There are countless situations where the information provided by sensors in an autonomous or semi-autonomous vehicle is not enough to avoid a collision; overall, in scenarios where the time to collision is minimal, and an evasive maneuver must be performed without delay in order to possibly safe the life of the occupants or that of pedestrians. Thanks to the incorporation of devices of diverse type, like: movement sensors, proximity sensors, GPS, LIDAR or video cameras, autonomous vehicles have the capability to reduce inter-vehicular distance to, among other things, minimize wind resistance and maximize the space used on the roads. However, the reduction of this distance implies that the minimal reaction time to an unexpected event is also reduced; therefore, it is required to implement measures to anticipate the possibility of danger and act opportunely. Currently, collision avoidance technology is based mostly in information received from on-board sensors; we consider that taking

advantage of communication capabilities, the vehicles on the roads can cooperate to execute collision avoidance maneuvers in situations where the reaction time is minimal due to the occurrence of an unforeseen event. In this research, we propose a new model of cognitive agent for collision avoidance that uses a hierarchical structure of fuzzy systems integrating information provided by a cooperative driving environment. The knowledge that the vehicle has about its environment, as well as its intention on the road, is critical for this model; thus, we are presenting an ontology structure to store it. We pretend to demonstrate that, in face of sudden events on the road, this model outperforms the reaction capabilities of a human driving a vehicle, and moreover the capabilities of autonomous vehicles performing avoidance maneuvers in an isolated manner. For that, as an additional contribution, we have developed a traffic simulator to visualize in 3D the cooperative vehicles executing the mentioned maneuvers.

LA CONDUITE COOPÉRATIVE POUR L'ÉVITEMENT DE COLLISIONS BASÉE SUR DES AGENTS COGNITIFS

RÉSUMÉ

Par : Giancarlo Colmenares

Les véhicules équipés peuvent percevoir leur environnement via senseurs à bord ; de plus, ceux qui ont des capacités de communication incorporées peuvent partager les informations reçues avec leurs voisins. Ces derniers, connus comme véhicules connectés, sont la base pour la conduite coopérative, un sujet d'intérêt dans le domaine des Systèmes de Transport Intelligents (*Intelligent Transportation Systems, ITS*) ; la conduite coopérative utilise les technologies de communication véhiculaire pour transmettre et recevoir des informations d'intérêt, telles que : le trafic, la sécurité, le routage. Cette recherche est axée sur la sécurité routière, plus spécifiquement nous sommes intéressés à la réduction du risque de collision des véhicules autonomes. Il y a des innombrables situations où les informations livrées par les senseurs des véhicules autonomes ou semi-autonomes ne sont pas suffisantes pour éviter une collision ; surtout, dans scénarios où le temps pour collision (*Time to Collision, TTC*) est minimal, et une manœuvre d'évitement doit être fournie sans délai pour possiblement sauver les vies des occupants ou des piétons. Grace à l'incorporation de dispositifs de divers types, tels que : senseurs de mouvement, senseurs de proximité, GPS, LIDAR ou caméras de vidéo, les véhicules autonomes ont la capacité de réduire la distance inter-véhiculaire pour, entre autres, minimiser la résistance au vent et maximiser l'utilisation de l'espace sur les routes. Cependant, la réduction de cette distance implique que le temps de réaction pour un évènement imprévu est aussi réduit ; par conséquent, il faut implémenter des mesures pour

anticiper la possibilité de danger et agir opportunément. Actuellement, la technologie d'évitement de collisions est basée principalement sur des informations reçues des senseurs à bord ; nous considérons qu'en profitant les capacités de communication, les véhicules sur la route peuvent coopérer pour exécuter des manœuvres d'évitement de collision en situations où le temps de réaction est minimal à cause de l'occurrence d'un évènement inattendu. Dans cet étude, nous proposons un nouveau modèle d'agent cognitif pour l'évitement de collisions qui utilise une structure hiérarchique de systèmes diffuses en intégrant des informations fournies par un environnement de conduite coopérative. La connaissance que le véhicule a reliée à son environnement, et aussi son intention sur la route, sont cruciales pour ce modèle ; donc, nous présentons une structure d'ontologie pour le stocker. Nous prétendons démontrer que, en face d'un événement soudain sur la route, ce modèle surpasse les capacités de réaction d'un humain qui conduit un véhicule et, de plus, les capacités des véhicules autonomes qui exécutent des manœuvres d'évitement de façon isolée. Pour le faire, comme une contribution additionnelle, nous avons développé un simulateur de trafic qui permet de visualiser en 3D les véhicules coopératifs lorsqu'ils réalisent les manœuvres mentionnées.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

Introduction

Autonomous vehicles have become a reality; several motor companies are including new modules inside cars to automatically take control in order to help in the driving task. Some of their features include: automatic parallel parking, automatic breaking in case of detection of obstacles in front, or course correction in case of detection of vehicles on the blind spot while performing a lane changing maneuver.

Progressively, vehicles have been released with new integrated characteristics aimed to simplify driving; some of them automatically perform certain tasks that were before an exclusive responsibility of the driver, e.g.: keeping acceleration, braking, controlling the steering wheel. Others are conceived to make a more pleasant ride, such as localization through GPS, finding route plans, or searching useful and updated information for the occupants. The introduction of communication technologies into motor vehicles will allow, in the near future, the exchange of data between cars through the creation of VANET (Vehicle Ad-hoc Networks) with direct vehicle to vehicle communication, and also with connection to specialized infrastructure in order to obtain traffic and weather data online.

Intelligent Transportation Systems, or ITS [1], [2], group a number of information technologies conceived to minimize and manage emergencies on the road, as well as those capable of providing useful information to the occupants [3]. Communication technologies, in particular, give cars the capacity to share information with others in the surroundings; although this information might be useful for a driver trying to avoid a traffic jam, it would be even more useful if it can allow several vehicles in a determined zone to cooperate and prevent a dangerous situation. Thus, this opens an array of research possibilities for vehicle networks.

As will be explained in detail in this work, cooperative driving is based on a collective behavior of a set of connected vehicles, more particularly those who are autonomous (without drivers) with coordination capacities. Inter-vehicle communications are an essential technology to cooperative driving [4]; some applications of cooperative driving are used to guarantee road safety, by using collected information from other vehicles to assist the driver in dangerous circumstances, or to warn about possible hazards on the road. The coordination of several vehicles involves sharing current car status and road information, which can include rules and specific limitations of each portion of the streets used by these vehicles. In a cooperative driving environment, a collective decision-making process is performed; several nearby vehicles can determine and perform joint actions in a secure fashion.

A vehicle equipped with sensors can detect the imminent occurrence of a collision, given current environment conditions; current developments are still in an early stage, they can produce visual or audible warnings for the driver, or gradually apply the brakes. Next phase in the evolution of this technology must be able to compute a matching evasive maneuver. Moreover, this type of maneuver has to include specific values of control variables, such as: acceleration, braking, and steering wheel angle at all times in order to safely and timely avoid the collision. In a cooperative driving environment, the computed maneuver must match evasive maneuvers of surrounding vehicles; thus, even if maneuvers are independently performed, jointly, they serve to a common global goal, which is to avoid the collision.

Before these types of collaborative solutions can see the light on the roads, they must be tested in several scenario configurations to evaluate their performance. The conception of the mentioned components, as well as other solutions for vehicular network systems is fundamentally based on the use of discrete event simulators [5], [6], [7], [8]. These tools are basic, not only for testing purposes of protocols and wireless communication technologies, but also to design and observe vehicles' behavior in multiple situations. This phase is crucial before testing and implementing in real life cars. So, simulation plays a very important paper, because it does not require to spend money to build equipped infrastructure or technology components; also, researchers can know in advance if a proposition is feasible or not. Therefore, simulations are a realistic, useful and economic solution to test and configure technological propositions generated in the scientific community for vehicular environments.

As will be exposed in the following chapters, current development of collision avoidance solutions based on cooperative driving is quite limited; there is also a lack of the realism in proposed solutions because they mostly use 2D simulations and do not integrate full vehicle control variables. Many assumptions are considered by researchers, such as: full coverage of connected vehicles, non-present or static physics variables (mass, friction, etc.), homogeneous characteristics of vehicles, among others that this research intends to tackle. Moreover, cooperative approaches for vehicle collision avoidance are mainly centralized, which gives no option to negotiation between actors and can produce a bottleneck effect when the number of vehicle increases; there is a need to explore distributed methods to cooperatively deal with collision scenarios, and to resolve conflicting plans. Street situations in the real life have variables that create unforeseen events while vehicles drive; snowy or slippery roads, fog or rain that block visibility are examples of problematic configurations that must be considered in order to produce realistic results for future application developments in autonomous vehicles.

Additionally, as vehicles are made by multiple vendors and the software on them is produced by different developers, it is required that all vehicles have a common understanding of the environment around them; information shared by vehicles in a VANET must be written in an unique format so that all agree on what is their current location, orientation and speed, what is the current status of the road, and more importantly what are the intentions of each vehicle when they face a collision situation. A cognitive approach that uses an ontology knowledge base is proposed to explore this challenge.

In the near future, connected and non-connected vehicles will share roads; they will find themselves sharing time and space in traffic jams, or in face of unforeseen situations that put them in danger. Having this in mind, the present work presents the problem of collision avoidance in cooperative scenarios, makes a study of current literature related to field, and proposes an innovative solution based on the use of cognitive agents; through the use of simulation, this research tests and validates a cognitive approach to create collision avoidance solutions integrating cooperative driving technologies. In this way, an original method is proposed in the research of the ITS field; this study is an additional component of the emerging smart cities, and its results can be reflected in more pleasant rides and in more saved human lives.

Chapter II

Context and current state of the problem

2.1 Introduction

Communication technologies have conquered a large part of human daily activities; from the way we work, share information, keep the agenda, study or enjoy our hobbies. Some of those tasks are especially complex and can only be performed by humans, e.g.: driving a vehicle. Initially, driver assistance systems incorporated cruise control capabilities; however, in this decade, a technological revolution has invaded vehicles, by integrating devices for localization, obstacle detection and collision pre-detection, just to name a few. More recently, new on-board capabilities can autonomously or semi-autonomously perform their work. Even if it sounds too soon, big car companies have estimated that for the year 2020 they will be commercializing completely autonomous vehicles [9].

Intelligent Transportation Systems (ITS) comprise groups of technologies conceived to minimize and manage dangerous situations, traffic accidents, and to provide useful information to the drivers [10], [11], [12]. Moreover, the study of technologies associated to the integration of autonomy characteristics in vehicles is also framed in the area of the ITS [10]. One of the focus of ITS is the conception of applications for road safety [10], among which we can mention collision detection and avoidance systems. As will be discussed in detail in this chapter, current on-board technology in vehicles uses sensors to detect its environment and recognize situations of risk; also, it is possible for some of such systems to warn the driver or to execute course correction maneuvers on behalf of the driver. However, there is a number of complex scenarios where the responsibility cannot be left to the driver, or where slightly applying the brakes is not enough to avoid a collision. Such scenarios usually include unanticipated events on the road, which consequently involve very little time to react, often impossible for a human to perform; a pedestrian that unexpectedly enters the street or a vehicle in front that suddenly applies the breaks are example of them.

This chapter is organized as follows: a brief presentation of the type of common road configurations for collision scenarios; then, an introduction to connected vehicles, with an overview of the technologies that make possible vehicle communications as well as its applications; later, we present autonomous vehicles and cooperative driving. A series of challenging scenarios that exemplify situations that can arise on the roads and which could benefit from further research on cooperative collision avoidance approaches are described on this chapter, they serve as the basis to justify this research work.

2.2 Problem scenarios

According to a research note of the U.S. Department of Transportation [13], there were 431,000 injured people in the U.S. in 2014 due to distracted drivers, also ten percent of fatal crashes involved lack of attention to the road. In Canada, the government of Alberta in 2015 [14], expressed that driver distractions correspond to 20 to 30% of car accidents, and that distracted drivers are three times more likely to be involved in a crash. Distraction is commonly associated to the use of the cell phone, but it can be also related to eating or adjusting the radio and climate controls. Given this situation, it is clear that technology can help reduce the frequency of accidents related to lack of attention; an autonomous vehicle can take control of the pedals and the wheels in order to stop itself or to correct course. Moreover, it can support the driver in situations related to failure on recognizing the status of the road, the capacities of the vehicle or miscalculating a driving maneuver.

Vehicles are more susceptible to be in danger of collision when the road infrastructure changes, when it is divided from a single way to several, when two lanes merge or when multiple lanes encounter at an intersection; virtually, at any point where the road changes from a single way configuration. When vehicles approach an intersection, they must reduce speed in order to avoid a collision with incoming vehicles from other lanes. However, there is a number of variables that enter to play here and can produce an accident; for instance, using Figure 2.1 as reference, we can see that: if *c1* decelerates too fast, the vehicle *c2* could have not enough time to stop opportunely because its maximal deceleration is not enough to keep a safe distance (*d*) between them. That can happen if the speed of *c2* is too high or if the driver of *c2* is not attentive and realizes too late about the speed reduction. Such circumstances can arise in other configurations, like in a highway when a vehicle wants to exit from it, and even involving more than two cars.

So, there is a problem associated to how vehicles behave when they encounter changes in the road infrastructure. Another type of hazardous situation occurs when a driving maneuver involves a blind spot, examples of this are a lane changing maneuver, a parking maneuver or any movement that requires backward motion. Like it can be observed in Figure 2.2, a driver trying to change lanes in a street might produce an accident because he/she cannot see the car in the blind spot. The same can happen with children or small objects on the road that scape from the view of the driver.

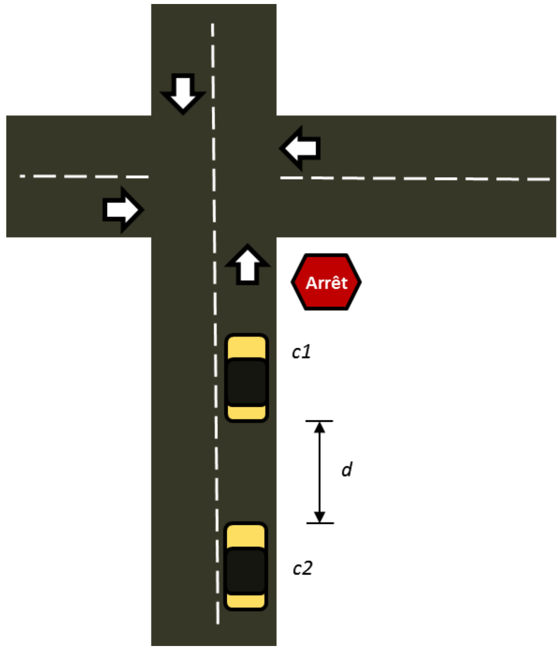

Figure 2.1 Rear-end collision scenario

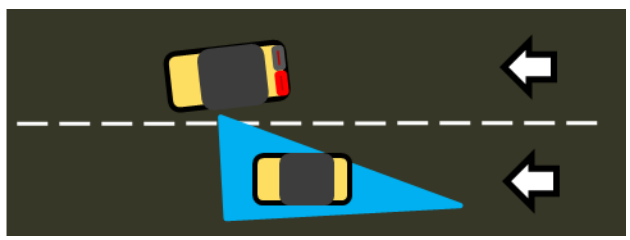

Figure 2.2 Blind spot in change of lane maneuver

To understand some of the variables that influence the occurrence of car accidents, we can mention various studies that explore the relationship between congestion and accidents; for instance, Wang et al. [15] have observed that traffic congestion has no impact in the frequency of accidents, Chang et al. [16] discovered that in fact the number of accidents tends to decrease when the traffic volume increases, which corresponds to a report of the statistical office of the European Union [17]. According to the report, this occurs because high traffic volume reduces average speed;

collaterally, when accidents do happen, the likelihood of fatalities is less due to the low speeds. Results of Retting et al. [18] show that the road infrastructure has a real impact in the number of accidents occurred in urban areas; he explains, for instance, that roundabouts tend to reduce injury crashes by 76% when replacing conventional intersections. Other interesting statistics, provided by a report of the U.S. Department of Transportation in the year 2008 [19], state that 30% of crashes involve one vehicle and almost 60% involve two vehicles; also, only 3% of the accidents occur in single lane streets, 45% and 52% occurred in streets with two and three lanes, respectively. About the configuration of the accidents, the report states that almost 27% were due to a change of traffic way or a vehicle turning, while 21% involved vehicles traveling in the same way and same direction.

Manufacturers have introduced elements of passive safety into the vehicles, which, as stated by Falcone et al. [20], are components primarily focused on the structural integrity of the vehicle, they also intend to minimize the effects of an accident over the occupants; examples of them are the seatbelts and airbags. Same authors explain that active safety systems are used to avoid accidents from happening, and also to simplify vehicle control in emergency situations; examples of that are anti-block systems (ABS) and traction control systems.

Therefore, we are interested in studying more complex situations in which current active safety systems are not enough to avoid the danger; particularly, we want to explore how more advanced technologies can help in the described scenarios as well as in others that will be presented. For that purpose, we will introduce connected vehicles, automated vehicles and cooperative driving, as we believe that such technologies can produce good results in collaborative environments, taking advantage of communication and automation capabilities.

2.3 Connected vehicles

2.3.1 Introduction

The concept of connected vehicles defines a type of car having communication capabilities, with other vehicles (Vehicle to Vehicle, V2V) or with equipped infrastructure (Vehicle to Infrastructure, V2I) [10], [21], [22], [23]; purposes of this communication are very diverse: road safety [10], [22], traffic minimization [10], [19], accident avoidance [10], efficient fuel consumption [10], [21], [24], navigation systems [22], or, more recently, for collaboration [21], [25]. Such wireless communication can be performed via sensors [10], [21], [23], radio signals [22], cellular communication [25], via internet and the IP protocol [10], [25], or via the Dedicated Short Range Communication technology (DSRC) [26], which was specifically designed for vehicle communications. Therefore, vehicles have now two new possibilities: a) to obtain disseminated sensor information about their environment, near or distant, depending on communication technology ranges, and make decisions based on it; and b) to propagate information they have about other vehicles, whether directly obtained or received from another actor.

The introduction of new equipment into vehicles helps the drivers to perform an easier and safer driving; different sensors and available devices give enough information to know where the vehicle is, if it is too near of another one, if there is vehicular congestion in the planned route, if the vehicle needs to change lane, remaining time to destination, among several others. Some manufacturers have introduced in their vehicles the Adaptive Cruise Control technology (ACC), which can automatically keep current speed and adjust it if the front car reduces its own, leaving the driver on control of the steering wheel; as it can be inferred, this technology can also support the type of problem scenarios presented in the previous section, as it can reduce speed to avoid a rear-end collision.

As the goal is to make a safer driving, technology has given steps in two senses, the first: give to the drivers the required quality information so they can make adequate decisions, i.e. driver assistance. And the second case consists in removing some (or all) of the driver responsibilities, and give them to the vehicle itself, i.e. autonomous vehicles. Desjardins and Chaib-draa [27] explain that advances in technologies related to inter-vehicle communication (IVC) support the conception of those driving assistance systems; moreover, they give a classification for this assistance: if it can take control of some tasks and it is performed without human intervention, it is denominated autonomous assistance, otherwise, it is semi-autonomous assistance.

Constant creation of new devices with environment sensing and detecting capabilities has allowed the integration of new functionalities inside cars in order to facilitate the driving task. Among these functionalities, are worth to mention: localization with GPS, and the detection of vehicles that are outside the driver's visibility thanks to proximity sensors; furthermore, video cameras, radars, speed controllers, cellular communication, traffic and map databases, among many others, help give more autonomy to vehicles [23]. To regulate and manage advances in this area, the US Transport Department created at the beginning of the 2000 decade the Vehicle Infrastructure Integration initiative, renamed in 2009 as *Intellidrive*.

Intellidrive is a group of technologies and applications that uses wireless communication to perform connectivity with better security, mobility and improvement on surface transportation environments [28]. The continuous integration of *Intellidrive* applications to vehicles, and the research on new possibilities, will help have safer and more efficient roads and transportations in the near future [22].

Around the world, advances in the implementation of this type of networks are performed as part of ITS; Drive C2X, Heero, Ecomove, Freilot and Instant Mobility are examples of projects using V2V and V2I, in the European Union, for intelligent and cooperative transportation. Similar projects or programs are developed in major cities of the US [29], Japan and Australia.

2.3.2 Communication technologies

Vehicle Ad-hoc Networks (VANETs) are a type of mobile ad-hoc network based on Vehicle to Vehicle communication (V2V) or Vehicle to Infrastructure communication (V2I) [30]. This type of network, unlike others, has a highly changeable topology and frequent disconnections. Nevertheless, as the nodes are vehicles, they are capable of providing enough energy and processor capacities, as well as constant communication with sensors inside the vehicle [31], which is an advantage over other types of networks. Inter-vehicular communication allows vehicles to exchange information about their environment, their speed and orientation [32].

Currently, it is common to find vehicles with GPS and navigation systems; however, they are limited to the street map residing in its internal memory. A connected vehicle, with a different navigation system, could benefit from information arriving in real time regarding recent events produced on the road [33]; thus, changes in the current planned route can be performed on the move. Remote information retrieval is possible even in the case when there is not direct connectivity with network infrastructure on the road because neighbor vehicles can be used for multi-hop communication [34], [35].

The requirements of a connected vehicle can vary depending on the installed applications [36]; in any case, they are commonly related to:

- Radio communication.
- Networking.
- Vehicle positioning.
- Other sensors and radars.

More and more, new sensors come installed inside vehicles to manage electronic systems and other car components, e.g.: rain, speed, acceleration, brake, friction, temperature and type of road detectors [29]. The most widely used communication technology is DSRC, for a great number of ITS services but especially for security information transmission [37]. Although, some researches on inter-vehicular communication propose to use wireless standards, such as: UMTS, WiMax, WiFi 802.11x [38], CDMA or TDMA [39].

The standard IEEE 802.11p, based on, and very similar to, the IEEE 802.11a, has some adaptations specific for the communication between rapidly moving nodes. The main difference between the physical layers of these standards is that the 802.11p uses the 10Mhz bandwidth. It is integrated to the IEEE P1609 Wireless Access for Vehicular Environments (WAVE) for the specification of the operation of the higher layers [40].

2.3.3 Applications for VANET

As was mentioned, connected vehicles can have additional components of various kinds:

- Proximity sensors: used to detect objects in the vehicle's vicinity.
- Cameras: these are used to recognize shapes and colors in the environment.

- GPS: used to precisely know the location of the vehicle.
- Communication capabilities: these allow the exchange of information with other vehicles in the network. Shared information can be divided into:
    - Vehicle status data, e.g.: speed, orientation, and acceleration or deceleration levels.
    - Data obtained from vehicle's sensors, e.g.: position of nearby objects, presence of pedestrians or traffic signals, conditions of the street, weather, etc.
    - Information calculated by the vehicle using a combination of the previous two, e.g.: the imminence of a collision, traffic jam formation or accumulation.

According to the classification presented in [32], applications for connected vehicles can be considered as one of two types: Intelligent Transportation Applications (ITA) or Comfort Applications. The former are those applications used to manage and analyze the traffic and the navigation; while the latter are those applications that improve the occupants' comfort by providing content of interest, downloaded from the internet or from other cars. A third class, or maybe a sub-group of the ITA, are the safety applications; those that alert about possible collisions or obstacles on the road, assist in a lane changing maneuver, or serve to the cooperative driving [41]. Willke et al. [42] also mention three types of applications of cooperative driving support systems: traffic safety, traffic efficiency and added value services; authors explain that with IVC, vehicles can obtain information coming from far away areas, which can be exploited by non-autonomous vehicles to opportunely inform drivers so they can act in consequence. Karagiannis et al. [36] provide a very similar classification:

- Road safety applications. To manage the safety inside a vehicle and on the roads with sensors [10]; pre-detection of collisions and to alert the loss of control [36].

- Traffic management. To manage the traffic with information obtained via ad-hoc networks or centralized systems [10], [28], [29]; traffic light control [25]; adaptive cruise control [25], [43]; lane changing [44], [45]; intersection management [21].
- Infotainment. Point of interest notification, download of multimedia content [36].

This research is particularly interested in studying safety applications that make use of cooperative driving technologies. We will be exploring solutions to detect the risk of collision situations on the road, as well as avoiding them, based on the integration of information coming from several sources; thus, we consider the present work can be framed in the road safety research domain.

The goal of road safety applications is to reduce the probability of accidents. According to Karagiannis et al. [36], most of the accidents are related to lateral, frontal collisions or to intersection crossing. Road safety applications aid to minimize the occurrence of this kind of collisions by providing useful information to the drivers. This information can be obtained from other vehicles and it can include location, speed and orientation in relation to an intersection; transmitted information can also allow distant vehicles to know the conditions of the road and the location of dangerous zones for the transit.

Mainly, three types of road safety applications, based on cooperative driving, are worth to mention:

- Warning: using information provided by neighbouring vehicles, an imminent collision or the risk of a collision can be detected, an emergency vehicle approaching, loss of control of a nearby vehicle, and dangerous road zones.

- Assistance: several vehicles sharing the streets can create coordinated plans to change lanes or to overtake slower cars; also, cooperative driving applications are useful to safely synchronize the convergence of several vehicles towards one same lane.
- Collision avoidance (CA): this type of applications is a combination of the previous ones; CA applications can foresee the occurrence of a collision, given current state of the environment, which is a fusion of information coming from several neighbouring vehicles. Later, they must create an avoidance plan that can integrate maneuvers of all vehicles involved in the collision; share this plan, agree on it and execute it.

2.4 Autonomous vehicles

Connected vehicles can be classified as autonomous or non-autonomous; this classification is directly related to the importance of the responsibilities the vehicle has while driving. Thus, non-autonomous vehicles can be defined as those limited to inform the drivers about events or situations detected on the road; while autonomous vehicles, also called driverless cars, are those capable of moving from one point to another without human intervention [46], sharing the roads with other vehicles (autonomous or not). Baig [46] depicts that the autonomy of these vehicles depends on their capacity to understand the environment via sensors. In addition, Shaikh and Krishnan [47] name a third, intermediate, classification: semi-autonomous vehicles; according to the authors, in these vehicles, the human is in control though there are autonomy components conceived to enable a safer control.

Even though most authors agree that an autonomous vehicle has the capacity to safely drive without a human, some others clarify that the autonomy concept also means that the vehicle does not communicate with other vehicles or the infrastructure [48]; yet others make a distinction between connected or not connected [44].

The fact remains that non-autonomous connected vehicles are also a fundamental part of Intelligent Transportation Systems; the main goal of this technology is to give information to the drivers in order to help them in the decision-making process [10]. Communication helps vehicles to be more efficient; thanks to the cooperation with other vehicles and with the equipped infrastructure, they can improve traffic flow and safety [48].

To integrate autonomy characteristics, cars have computers, LIDAR or cameras, and global positioning systems [23]. These devices provide data of the near environment to the vehicle to help it locate itself on the street, detect other vehicles and objects on the road, and also to recognize the infrastructure. Hu et al. [44] mention three main characteristics related to autonomous connected vehicles:

- There is precise traffic information available online.
- Ultra-short reaction time.
- Cooperative driving.

To prevent accidents produced by not noticing a car in a blind spot, as illustrated in Figure 2.2, manufacturers have implemented at least three technologies: blind spot detection (BSD), lane departure and parking assist. BSD systems are currently used by Ford, Lincoln and Volvo, just to name a few; they use ultrasonic or radar sensors on the sides and the back of the car to track objects in the blind spots, when an object is detected the system can either trigger an audible alert, flash an indicator in the mirror, or even make a light shake of the steering wheel if the turn signal is turned on. The lane departure warning systems use a forward-facing camera to track whether the car is centered on the lane or not, if the car is going out of lane it triggers an audible or a visual alert on the dashboard. A similar combination of technologies is used by parking assisting systems, they have a rear-facing camera and show the real-time video on the dashboard for the driver, also they could have ultrasonic or electromagnetic sensors to indicate how close the car is to objects nearby. These solutions provide the necessary means for an autonomous vehicle to create a plan to avoid the collision; however, they are currently limited to warn the driver and are not capable of executing an action.

An interesting situation arises when two vehicles are on the same lane of a street and the one behind is following from too close; this scenario is depicted in Figure 2.3. Assuming that vehicle *c1* is an autonomous vehicle and *c2* is driven by a human, then *c1* has to consider a number of variables in order to make a decision and avoid a rear-end collision. For instance, would it be better to accelerate or to change of lane? The decision on one or the other depends on the speed limit, the closeness of an intersection, the presence of another vehicle in front, or the status of the road, just to name a few; moreover, if there is an approaching vehicle on the next lane, is there enough time to make the lane change, or is it better to wait for it to pass?

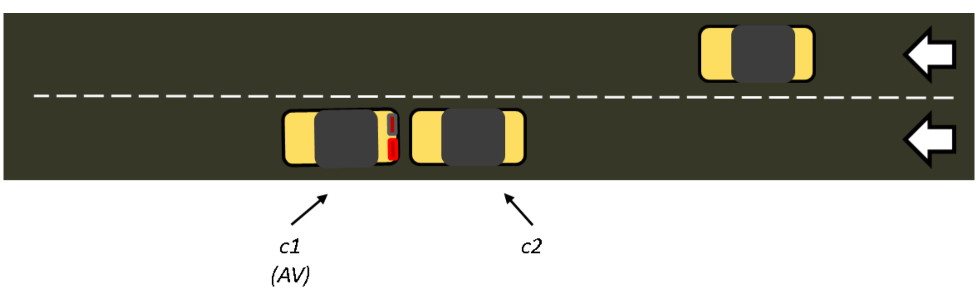

Figure 2.3 Non-connected car follows too close an AV

A similar set of considerations must be taken by an autonomous vehicle that finds bad conditions on the road, like rough pavement, or slippery due to ice or snow. In such case, a decision must be taken promptly in order to avoid the bad section and prevent an accident; however, in these new circumstances, acceleration is not a possibility. If both vehicles are autonomous, they could find a collective strategy that avoids the danger for them; which is one of the goals of the research in cooperative driving.

2.5 Cooperative driving

2.5.1 Introduction to cooperative driving

According to Terroso-Sáenz et al. [49], cooperative driving systems support security improvement and traffic management; which are based on wireless communication in order to dynamically exchange information among vehicles, particularly using Vehicle to Vehicle communication (V2V). In this sense, several objectives of cooperative driving are found, such as the detection of dangerous situations in real time, by the means of a diversity of sensors. Cooperative driving is also based on the integration of information coming from sensors in other vehicles [50].

Cooperative driving provides an environment for shared decision-making process between vehicles; which is a logic that organizes the behaviour of a group of vehicles based on the occurrence of events [51]. Coordinate joint actions to exit from a given situation during driving is one of the objectives of cooperative driving. The introduction of this technology is intended to create safer and more efficient driving environments; cooperative driving systems can use sensors and communication capabilities to reduce the distance between vehicles and to stabilize the traffic flow [52]. Car platoons [42], [52], [53], [54], [55], ordered intersection traversal [21], [56], [57] and speed synchronization (cooperative cruise control) [27], [43], [58], [59] are examples of possibilities provided by cooperative driving.

Therefore, cooperative driving can solve, conflicts and dangerous situations between vehicles, where the precision and computing level overcomes human capacities. There could be situations, like in the problem scenarios, where the number of variables to consider is to high, or impossible to know in advance for a human being; for instance: how slippery the road is or the speed of the car coming behind. If the time taken by the driver to recognize the danger, consider all possible future states of objects in the surrounds, and ponder every possible evasive action is greater than the time to collision, then the scenario will end in a tragedy.

2.5.2 Challenges of cooperative driving

Equipped vehicles forming a VANET must be able to make decisions based on information disseminated by other vehicles [32]; they should fusion location, speed and other data from several vehicles in order to be guided in a coordinated fashion, as members of a unique entity performing a task.

Hence, a data structure must be created to be used by vehicles, so they can share and integrate information; it will allow to have a common understanding of what is happening in the surroundings. The fusion of data from multiple sources requires that the actors, generating and consuming information, use a common language and to agree in a global and unique explanation of the situation.

Among the challenges of cooperative driving we also find the conception of applications to coordinate the order and speed at which vehicles cross an intersection. In highly frequented intersections, it is common to find traffic lights giving authorization to pass to incoming vehicles. However, using traffic lights, or stop signs, like the scenario illustrated in Figure 2.1, generates a delay because cars on some lanes have to stop to ensure safe crossing of others, this is translated into bigger traffic accumulations, increase of fuel consumption and of CO2 emissions. Vehicles have to decrease their speed to zero while approaching to a red light, which produces an unnecessary reduction in the average speed of a whole section of the traffic; similarly, when the light turns green, they must recover speed. Such configuration also requires an unnecessary consumption of time and energy if no vehicle passed on the other way.

The coordination of this kind of situation can be seen also as an optimization problem; adapted to scenarios covering entire sections of a city with hundreds of cars and several intersections. An important consideration to make is the goal: to maximize the global or the average speed, to minimize the global or the average waiting time; such decision will have different results and implications in the traffic system. Furthermore, computing time is a main concern, because an autonomous vehicle approaching to an intersection is driving, and at the same time it is waiting for indications on how and when to cross; then, a definition on a centralized or distributed scheme needs to be done as well.

Another challenge of cooperative driving systems is to safely guide several vehicles in a single lane, one behind the other [51]; the longitudinal distance among vehicles is very important in the driving strategy because with short distances air resistance is reduced and aerodynamic forces are lighter [60]; which results in a reduction of fuel consumption. In this sense, vehicles in such configuration with cooperative driving technology are continuously communicating data about their location, speed and other movement data; the idea is that vehicles use this information to coordinate an efficient autonomous driving by taking advantage of the aerodynamic.

Moreover, to make the most out of the aerodynamic, it is necessary to have a minimum inter-vehicular distance. Thus, vehicle location must be precisely computed, as well as the relative distance and speed with other cars. Optimal accomplishment of this goal is complex and dangerous; an algorithm that computes required acceleration forces, to attain the necessary speed and keep minimal proximity between moving vehicles, will have to consider minimal reaction time in front of unforeseen situations on the road. Hence, to ensure passengers' safety, this precision task cannot be held in human hands. According to Schito and Braghin [60], minimization of these distances can only be undertaken with the development of technologies for autonomous control of the vehicle's position. To have an idea of the problem, let us consider Figure 2.4; it shows five vehicles driving with minimal distances (*dmin*) between them. Assuming that *dmin* is 2 meters and vehicles have a speed of 30 km/h, if a person enters the street in front of the lead vehicle and it brakes, the second car in the platoon has 0.24 seconds to be aware that the first one slowed down, execute an internal decision-making process and finally apply the brakes; if vehicles go at 50 km/h and *dmin* is 1 meter, that time is reduced to 0.072 seconds.

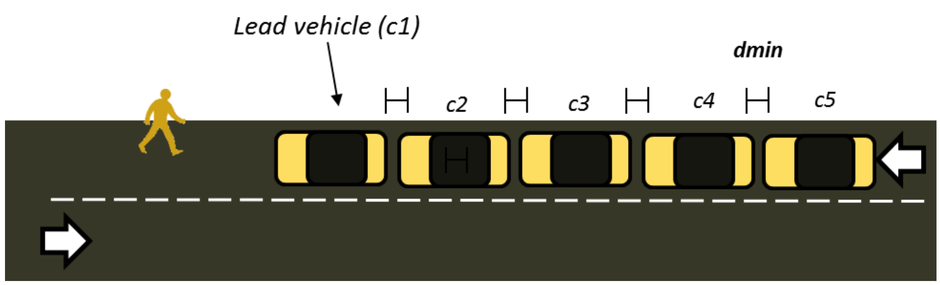

Figure 2.4 Cars driving one behind the other and a pedestrian enters to the street

Nowadays, thanks to the use of sensors, ACC is used by vehicles in order to keep a predefined space measured in time [27]. Nevertheless, this technology is somehow limited because its distance sensors are on the bumpers [60]; so, they can only capture information related to the vehicle in front and not that of those going beyond. Using such technology helps to detect when the vehicle in front is reducing speed; however, it would be more useful if, considering Figure 2.4, vehicle *c5* knows that the lead vehicle braked or started an evasive maneuver. It is a current challenge to opportunely provide necessary emergency information to vehicles in the vicinity, and also to use it when it arrives.

Thus, ACC's capacities can be extended by taking advantage of information about the status of preceding vehicles in order to reduce at minimum the distance between them [55]; so, the use of V2V is an added value that allows vehicles to take anticipated decisions about best speed and braking values, considering data of vehicles around. This is known as Cooperative Adaptive Cruise Control (CACC). As an advantage of this technology, Nieuwenhuijze et al. [53] point out that the computation of optimal acceleration and braking values is still a matter of study; finding the best values of these control variables, not only could help avoid an accident, like in the scenario of Figure 2.4, it also minimizes energy loss, which constitutes a lower effort to recover speed, thus reducing fuel consumption.

Another issue related to cooperative situations is how automated vehicles behave in heterogeneous scenarios, their performance can be reduced when non-automated intruder vehicles appear. It is a fact, that the integration of autonomous vehicles to the roads will be a gradual process; therefore, before having a fully autonomous environment in our cities, there will be a diversity of autonomous, semi-autonomous and non-autonomous vehicles sharing the roads. Moreover, the variety of algorithms, data structures, sensor and communication devices will be as vast as the number of car makers interested in this domain.

Cooperative intersection traversal algorithms must consider that cars without cooperative capabilities, or non-connected at all, will appear at an intersection; so, a traversal strategy has to include it, even if it is not an active part of the cooperation. In the same way, collision avoidance systems must be designed considering that not all vehicles, in a collision situation, have avoidance capabilities or not even communication; the detection of such un-cooperative components in a scenario is a key issue to be tackled (view Figure 2.3). Therefore, it is required that the algorithms to conceive expect such heterogeneous configurations and are able to find collision avoidance strategies for them; specially for the early days of the introduction of cooperative driving and autonomous vehicles technologies.

Cooperative cars in an heterogeneous environment can know and coordinate in advance how they will react to a dangerous situation; however, that is not the case for non-cooperative or non-connected cars driven by a human. The multiplicity of human reactions in front of a collision scenario is still a matter of study; at what moment the human realizes of the situation, what the human can do, what would be his/her reaction time, what error margin should be considered for his/her evasive maneuver, are just an example of the circumstances to consider and that could be different from one person to another, depending on a variety of variables, such as: time of the day, time on the wheel, age, gender, emotional status, etc.

2.5.3 Collision avoidance and communication

Currently, vehicles have incorporated collision avoidance mechanisms; commonly, they use proximity sensors located on the bumpers to detect the presence of objects. As previously mentioned, some systems inform the driver about the possibility of a collision, via lights on the dashboard or with audible alerts. More recent systems go further and are capable of applying preventive measures, e.g.: autonomously apply the brakes; it is the case of Volvo with the City Safety system, Mercedes-Benz with the Pre-Safe system, Toyota with Pre-Collision and Lane-Deviation systems, and Nissan with Intelligent Brake Assist, among many others with similar solutions. The Intersection Movement Assist (IMA) is another example of this kind of systems, as Maile et al. [61] explain, it uses the perception-reaction time (PRT) of the driver to compute the appropriate time to warn him about the imminence of a collision with a crossing car.

So, current technology integrated in vehicles is able to detect the imminence of a collision and try to avoid it by pressing the breaks. Nevertheless, this autonomy of the collision avoidance system requires a line of sight with the other object (a vehicle or not); also, it is assumed that the other object will continue with its current state (stationary or not). So far, these systems do not consider the possibility of collaboration in order to avoid a collision; which is a matter of high relevance because, not in all situations, the actions of a single vehicle will be sufficient to avoid a collision.

Furthermore, let us imagine a situation where two vehicles are in course of collision, just like in the scenarios presented in the first section of this chapter; both of them could be capable of detecting the danger and generating an action plan to avoid it. However, if these cars only use data provided by on-board sensors, the plans they will generate might involve the execution of evasive maneuvers that are conflictive with the other car's plan; and therefore, they could end up creating a new collision situation, derived from the previous one. Even if both cars are able to detect the new dangerous situation, they will have less time to react to it. So, information of what the car can "see" with its sensors, and what it can deduce the other car will do, is not enough to produce an effective collision avoidance maneuver.

The use of communication technologies to build an environment in which vehicles cooperate to avoid a collision, is the response to collision scenarios in which sensors installed in vehicles have not line of sight with the other vehicle. Therefore, more "intelligent" vehicles can benefit from this capacity to inform others about its presence, and even to send variables associated to its position, speed, direction, etc. In this sense, technology is in the research stage, with advances in the integration of systems for collision avoidance in intersections [61], [62], [63]. In the near future, applications will use V2V to share information, coming from intra-vehicular sensors, with vehicles in the vicinity, to calculate the imminence of a collision, and to create cooperative plans in which several vehicles act to avoid it.

As mentioned, cooperative driving systems rely on information provided by several groups in order to compute intersection crossing strategies or collision avoidance maneuvers; thus, it is of major importance that communication devices on vehicles are able to opportunely send location, speed, orientation and other status information, as well as to receive it from others. When a high number of vehicles are in a zone, the quantity of transferred messages could be superior to the capacity of the on-board equipment; so, there is a problem related to the dissemination of information that requires an exploration of a series of elements, like deciding what messages must be sent, what strategy use to send them, and the performance of the protocols used in the VANETs. However, cooperative driving systems must anticipate the possibility that such problems will occur, information might not arrive at the expected frequency or it might not come from every vehicle in the vicinity; how to deal with communication breaks, and still keep road safety, is still a challenge for this technology.

2.6 Summary

As exposed, current technology included in commercial vehicles is capable of assisting the driver in several tasks; for instance, by providing traffic information disseminated via the vehicular network, or by detecting dangerous situations on the road. Moreover, recent vehicles can autonomously start preventive measures to avoid a collision, or assist the driver to remain centered on the lane.

Even if current technology is able to detect dangerous situations and actively try to avoid them, there are still situations that are not covered. Some collision scenarios cannot be resolved by the action of a single vehicle; furthermore, actions of several vehicles, taken independently, can produce new dangerous situations. Thus, in this chapter we have considered more complex configurations that can end up in a collision if an avoidance maneuver is not executed on time; they are intended to illustrate that the maneuvers of a single vehicle might not effectively prevent an accident. In fact, it is possible that, maneuvers coordinately executed by several vehicles can be easier to perform and have better results than the isolated evasive actions of all involved cars.

Rear-end collisions take the attention of this research because they represent a relevant portion of accidents on the roads; such situations, with minimal reaction time to avoid the collision, are considered here as part of scenarios involving only autonomous vehicles or a heterogeneous configuration with non-autonomous cars.

As discussed in this chapter, we are also interested in the study of collision situations that involve a lane changing maneuver. The blind spot problem described reveals the necessity of collision avoidance solutions that do not rely only on on-board sensors. Similarly, since it involves more than one vehicle, it is a source to test cooperative methods that can produce improved solutions compared to those created individually.

Due to the fact that technology will not be adopted by all vehicles from one day to another, two possible situations come up: the gradual introduction of cooperative driving features for collision avoidance in the vehicles, or the introduction of fully collision-free vehicles. Thus, it is of particular importance to consider heterogeneous scenarios, in which different types of vehicles share the roads; the interaction between computer assisted vehicles (fully or not) and human driven ones will be one of the biggest challenges to approach.

Given that multiples vehicles are involved in the situations; it is essential to review cooperative approaches in the literature that can provide an important contribution to this study. Therefore, in order to explore the exposed problems, the present research envisions three main research axles: the study of autonomy control for vehicles, collision avoidance strategies in urban environments, and the cooperative resolution of plan conflicts. They will be reviewed in the next chapter.

Chapter III

State of the Art

3.1 Introduction

Even though there has been some progress to adopt autonomous driving solutions for motor vehicles; they, mostly, consider vehicles in controlled environments, with standardized or static variables, which do not reflect the variety of complex conditions that can occur while driving a car, and which can derive in hazardous situations for the occupants and for pedestrians. As was explained in the previous chapter, there is still a problem associated to the resolution of dangerous situations produced by unforeseen events that occur on the roads, such as: sudden appearance of a pedestrian, abrupt braking of the heading vehicle, or rough turning maneuvers of neighbouring vehicles; in the same way, lack of visibility or limited traction with the asphalt, are examples of weather considerations to take in mind in order to avoid collisions.

With the purpose of tackling this problem, in this chapter we present literature related to the research axes of interest of the present work. The objective is to explore the state of the art and be aware of the fundamental elements needed to build a realistic model for an autonomous vehicle; also, to know the advances in the study of cognitive solutions for the development of intelligent agents. Therefore, the chapter is organized as follows: approaches associated to the physics of a vehicle are exposed, the organization of its status variables and the driving control; later, we consider the required elements to give cognition to an agent's behavior; afterwards, we present works associated to modeling cooperative capacities for communicating agents; and finally, we discuss researches focused on the study and design of solutions for collision avoidance.

3.2 Vehicle architecture modeling

In order to simulate the behavior of a motor vehicle, it is necessary to create a model that represents the different aspects that regulate it. Given that this research intends to study cognitive solutions to control an autonomous vehicle while evading a collision, we review the state of the art in two modelling areas: the first one comprises the works that explore the characteristics of dimension, movement and physics of a vehicle; the second one includes those researches focused on defining the cognition of the vehicle, which refers to the functions that allow it to make a decision in front of the current environment situation. Therefore, considering these models, this section firstly presents the aspects of the physics behavior, and later considers current literature related to the cognitive behavior.

3.2.1 Physics behavior

Based on the vehicle's kinematics (view appendix A for details on kinematics), a steering control method must be used to guide the vehicle over a known path. It must determine, at all times, the required heading angle to accurately track the defined path [64]; in this way, the vehicle can be leaded while driving over a previously computed path. Two main strategies of path-following algorithms are found in the literature: Geometric methods and control techniques. In the first type, the pure pursuit method [65] and variations of it are the most commonly used; the idea of this approach is to use a target point on the path that must be chased by the vehicle, computing the angle between this point and current position of the vehicle guides the vehicle towards it, and eventually to the path.

Control techniques, such as non-linear approaches, are also popular because they give more robustness by considering further variables to minimize the cross-track error; nevertheless, they are computationally intensive [64], so they are preferred for more complex situations, e.g.:

Unmanned Aerial Vehicle (UAV) path-following considering wind disturbances and aerodynamic coefficients.

The method presented by Thrun et al. [66], used by an unmanned vehicle robot in the DARPA Grand Challenge, uses a combination of the cross-track error and heading error to steer the vehicle along a defined path. The controller is based on a non-linear feedback function, for which the authors have shown convergence. Initially, the method finds the closest point on the defined path from the center of the front axle; then, given current speed, the controller will return the steering wheel angle required to get these points closer. This approach is depicted in Figure 3.1.

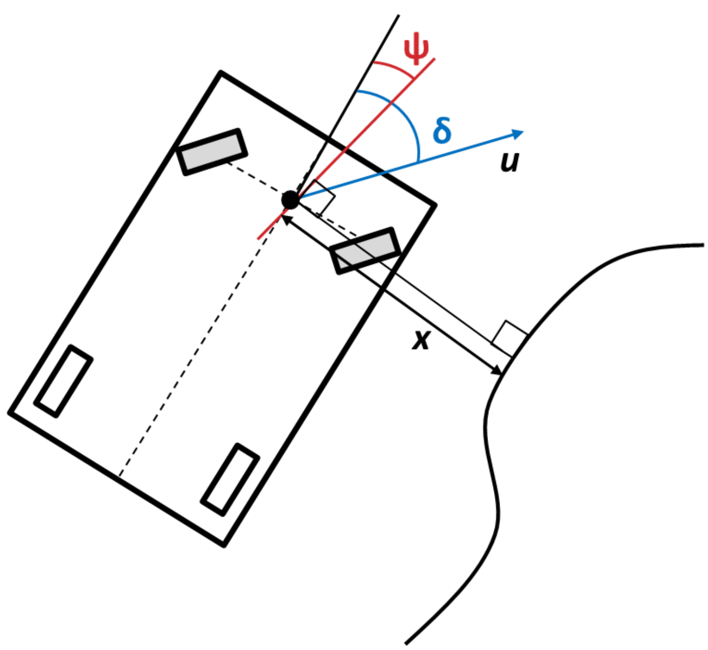

Figure 3.1 Thrun et al. [66] path following method's geometric description

Where, x is the cross-track error, u is the speed, ψ is the orientation of the nearest path segment, measured relative to the vehicle's orientation. At time t, the control function for the required steering angle (δ) is given by (3.1).

$$\delta(t) = \psi(t) + \arctan\left(\frac{k\, x(t)}{u(t)}\right) \tag{3.1}$$

Where, k is a gain parameter that determines the rate of the convergence. The work of Thrun et al. [66] has been a major contribution for researches in the field of autonomous cars and automatic control for driverless vehicles for the latest years [20], [67], [68], [69], [70].

The use of a model of this type can keep knowledge and control of kinematic variables of a vehicle moving on the streets. The simplification of the Ackerman model to a bicycle model minimizes de number of calculations and therefore, the time needed to compute a driving maneuver; which is a desired requirement for a collision avoidance capable vehicle. The kinematics serve to know the current status of the vehicle, e.g. width, length, orientation, wheel orientation, max wheel orientation, speed; however, some of them have to be modified in order to move the vehicle. Such modifications depend, at the same time, on speed and orientation variables; this is the role of the steering control, it updates vehicle's speed and orientation to lead it towards its planned trajectory, bounded by kinematics and physics constraints.

In addition to the physical behavior of the vehicle, a reasoning process recognizes current situation of the environment and generates a plan, which finally will derive in concrete actions of the pedals or the steering wheel. This reasoning process performs in a superior level model, known as the behavior model.

3.2.2 Cognitive behavior

This section of the chapter is organized to present researches related to an agent's cognition, as well as to consider the organization of the agent's architecture; together, they rule the complete agent’s cognitive behavior, from the reception of external input up to its processing to produce an action. Finally, architectural approaches created to model agents that control vehicles are also discussed.

3.2.2.1 Agent cognition

Poole et al. [71] explain that computational intelligence (CI) is a discipline of the cognitive sciences, and researches in this sense are focused in building machines that not only copy human behavior, but are also intelligent. In the same way, they mention that the language is inherently a symbol transmission, from the outside to the brain; thus, if the human reasoning, in terms of language, uses symbols as input and output, the reasoning functions could also be implemented in terms of symbol processing.

Even if some researches are focused in the use of black box approaches, such as neural networks, to emulate the reasoning process, one of the objectives of the CI is to understand the principles that rule that reasoning; thus, the trial to implement this functionality in a symbolic manner. The advantage of computational intelligence, or cognitive computation, is that it allows to study the knowledge and the intelligence, not just by observing its outer behavior, but by experimenting through models of intelligent behavior. Those are open to review, modification and experimentation. In [71], authors conclude stating that the final goal of CI is not necessarily to simulate in full scale the human intelligence; according to them, it is important to make the computer to solve a problem by reasoning as a human would do. Therefore, if it can give an explanation of how it resolved to find an answer, it can be considered as a form of cognitive modeling.

Furthermore, Castelfranchi [72] describes autonomy and cognition as features of an intelligent agent. He explains that autonomy is a relational concept, because the agent is autonomous in relation to something else (another agent, the environment, etc.). A cognitive agent has interpretations and meanings, in other words, it has *beliefs*; consequently, the agent is able to react to those beliefs. Reactions of an agent are simply behavioral patterns; these can be described in the form of goals, which are a representation of a state to which the agent wants to reach. Thus, a goal guides the cognitive behavior of the agent; there exists a *mental representation* between the external stimuli and the response of the agent, which is used and manipulated by its cognitive capacity.

As exposed before, a vehicle having semi-autonomous or autonomous capabilities has to have the means to input data of what is happening in its surroundings, and incorporate it to its own knowledge in order to produce an expected behavior. Thus, it can be seen as an intelligent agent; one that knows and has an understanding of its environment, thanks to its sensors, and acts over it via the actuators.

Burmeister et al. [73] describe an agent model for an autonomous vehicle, one that provides the means to input data of what is happening in its surroundings, and integrates it to its own knowledge to produce an expected behavior. It is composed of a series of modules (Figure 3.2):

- Actuators: they are in charge of performing the actions in the environment.
- Sensors: they are in charge of perceiving the environment.
- Communication: it is in charge of the communication with other agents.
- Motivation: it models the long-term objectives.

- Cognition: it controls and verifies the agent's individual, communicative and cooperative activities.

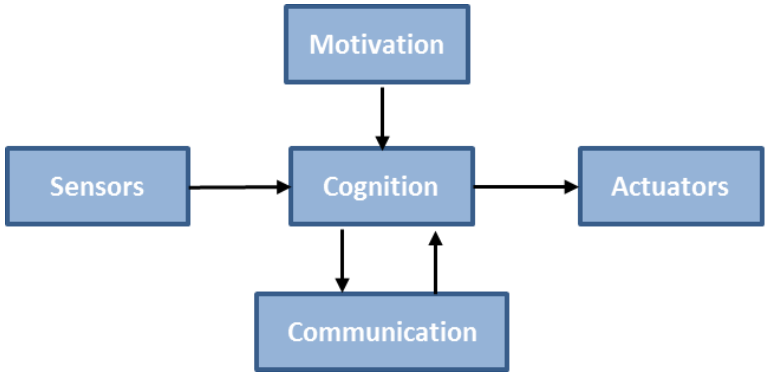

Figure 3.2 COSY architecture [73]

While sensors and actuators give a vehicle the possibility to see and act over the environment, it is of a particular interest the use of communication capabilities; why should an autonomous vehicle need to communicate? As it was presented in chapter 2, connected vehicles can share information with other vehicles or even with equipped infrastructure; and the idea is to have more information than that available just by using sensors. The COSY architecture [73] provides a simple structure that clearly divides responsibilities between the modules; it is of particular interest the connection of the communication module to the cognition one, because it is considered as a process that works in parallel of the agent's motivation and integrates with them. In this way, the agent's beliefs and intentions can be shared and integrated with those of other agents as part of an integrated process, and not just for information purposes. Sharing and using shared information is the basis for the cooperative behavior and it will be detailed later in this chapter. In this section, the role of the cognitive behavior and how it can be of utility when designing cooperative solutions, is presented.

Going further, a cognitive agent can be considered as one having the capacity to obtain knowledge from the outside, understand it, integrate it with its own previous knowledge and use it to drive itself towards the completion of its goals. Such knowledge can be considered as a mental representation of the world it is on, and therefore used as the core of the agent's cognition. A significant contribution of such capacity is that it can allow the agent to react differently when it faces similar collision scenarios with diverse environment conditions.

Knowledge, defined by Poole et al. [71], is information about some domain, and it is required to perform tasks related to it. So, in order to have a computational component that uses and reasons with that knowledge, it needs a representation and reasoning system. This system must specify a language, a way to give meaning to the language and procedures to create answers to inputs in that language.

One way to express knowledge is to describe the world in terms of individuals and relations between them. Then, the ability of a cognitive agent to solve problems in a specific domain is directly associated to how is that domain expressed through individuals and relations. In relation to this topic, authors in [71] state that an agent's cognition comprises four interacting tasks: modeling the environment, evidential reasoning, action and learning from experience. To the matters of the first task, Regele [74] proposes an ontology based world model to support Intelligent Transportation Systems applications, more specifically, autonomous vehicles. This model has two levels of abstraction, a low level with more grained and raw information about the environment, such as: sensor data and geometric values; in the higher level, more abstract information is held, like: relation between objects, such as street signs and traffic lights, and behavior interpretation. He explains that for the case of traffic coordination in intelligent locations, it is more important the chronological position of the vehicles than their spatial position. This model uses relations between lanes of the streets in a semantic context expressed in the form of 3-tuples: T = (L, O, R), where L, O and R, are finite sets of lane sections, lane objects and lane relations, respectively.

Such model, provides a vehicle with the possibility to decide if a lane changing or an overtaking maneuver is safe, by inferring on the relations and the facts stated in the ontology. The abstraction of the model minimizes the computing of geometric possibilities and the logic steps provided as solutions ensure their validity.

Patron et al. [75], on their side explain that situation awareness is the capacity to understand and deal with the environment, and they divide it into three levels: perception of the environment, comprehension of the situation and projection of the future status. This is a similar way to consider the steps of cognition of an agent; one that reflects how the agent expects to be in the future, given current its current state and that of the elements around it. Therefore, actions can be generated as the steps needed to pass from current state to the future one. As an important consequence of an increase in agent's awareness and control over the vehicle, authors explain that it reduces human awareness on that particular task, giving him the opportunity to focus on other tasks.

In the same sense of context, Ejigu et al. [76] present a definition for it as: "*Any information that can be used to characterize the situation of an entity*", and possible entities being a person, a program, a car, or any object (real or not). They explain that the emergence of pervasive and ubiquitous computing has as a consequence the need of devices that are aware of their context; such devices must be able to behave accordingly to the context they are on. So, they propose an ontology based model for context reasoning. Basically, they define contexts, rules and their semantics as triplets in OWL; by the means of an inference engine, context aware devices take decisions for the user without his/her intervention. This particular research is limited to change the status of a cellphone according to its location, considering a time constraint; it is directly related to the term of chronological position, described in [74]. Such term can also be used as a way to describe vehicle collisions, due to the fact that they are chronological events as well, involving more than one vehicle at a single location, and at the same time instant,

Chen et al. [77] also use an ontology to give context awareness to systems. They describe a Context Broker Architecture (CoBrA) that uses OWL to model basic concepts of people, agents, places and events; and, an ontology inference engine to reason over the semantics of the ontology. This architecture provides a platform for controlling and triggering devices' actions, based on location related knowledge. As a drawback, the authors have not considered temporal components for their research; however, the proposal provides a basis for the interoperability of independently developed solutions.

Wannous et al. [78] introduces a temporal ontology to model semantic trajectories in their research. Concepts for time, date, interval and related, are used in combination with rules for *before, after, overlaps, during,* etc. to manipulate and reason over trajectory data. Similarly, the Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) is introduced by Chen et al. [79], it integrates modular vocabularies in Web Ontology Language (OWL) to represent time, space, events and similar concepts. Authors of this research explain that intelligent pervasive systems can be modeled as intelligent agents; thus, the ontology also includes entities for agents having knowledge, belief, intentions and obligations. Other components that are part of the agent mental state, such as: goals and plans, are as well considered in the ontology. Particularly, the inclusion of spatial and temporal statements in a knowledge base, is a significant contribution to describe street events in a standard format; the integration of such feature can be a foundation basis to build safe applications for Intelligent Transportation Systems. More details and experiments with SOUPA can be found in [80].

Cognitive behavior directs the capacities of an intelligent agent, how it saves information of its environment and how it interprets this knowledge to react with actions; at the same time, an architecture expresses how information is received, processed by one or several behaviours, and sent out from the system. The organization of behaviors and the order of steps to process inputs coming from sensors, and produce outputs through actuators, is the responsibility of an intelligent agent architecture.

3.2.2.2 Agent's architecture

Multi-agent architectures, according to Huhhns and Stephens [81], allow modeling intelligent systems made up of several components, called agents; they have a behavior that is more or less intelligent. These agents are able to communicate according to some rules defined in the architecture with the goal of achieving independent or common objectives. In this section, architectures conceived to design the reasoning behavior of agents are presented; they allow the accomplishment of objectives by defining them in the form of sub-tasks that can be achieved independently.

Internal activities that rule an agent in its decision-making process can be organized in a serial or a parallel execution. This arrangement has derived in a general classification of layers, horizontally or vertically distributed. In the first type, layers have the same execution priority and each one is able to independently generate its own action proposal. In a vertical distribution, there is only one unique action generated by the layers' interaction, the internal process of each layer depends on the result of the previous one. These organizations have opposite advantages and disadvantages; a horizontal architecture has a processing overhead because all layers are activated to analyse the situation, while in a vertical one upper layers are called only if layers below are not capable of resolving current circumstances of the environment.

From the literature related to architectures of intelligent systems based on agents, we can observe two main categories: deliberative and reactive. Wooldridge and Jennings [82] explain that a deliberative architecture allows agents to use logic to make decisions, based on predefined rules, possibly written in a specific language conceived to this goal. In the other hand, a reactive architecture does not include symbols and does not reason through them; authors explain that some agents have behaviors and they act according to them to perform tasks and to assign priorities; while others are linked to groups of competences or abilities. The creation of agents whose decision-making process is based on rules, written in a common language, is advantageous because it allows them to easily share knowledge and find integrated solutions to conflicting plans; mainly because there is no need to translate the world model from one to another.

Wooldridge [83] states that the “Belief-desire-intention” (BDI) architecture is based on the reasoning principle to decide, at each moment, what are the actions to execute in order to accomplish the goals. Two important processes can be derived from here: to decide which are the objectives and how is the agent going to accomplish them. This architecture proposes that agent’s intentions lead it towards the actions that it has to perform; those intentions are based on current knowledge of the environment (beliefs) and on the group of options that the agent has to perform (desires). The knowledge of the environment refers to the information that the agent has regarding the environment it is working on. Using current environment information (it can change) and current agent’s intentions, it is possible to generate a group of available options to be executed by the agent, these are the agent’s desires. So, there is a cyclic relation, between the knowledge, the desires and the intentions of the agent, which guides the accomplishment of its objectives; intentions can change according to new events produced in the environment and to the consequent change in the agent's desires. Figure 3.3 depicts the reasoning process of an agent with the BDI structure.

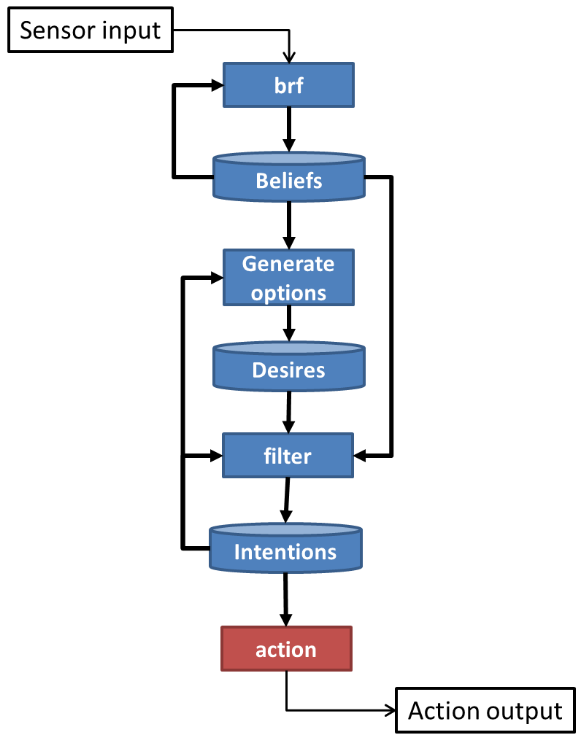

Figure 3.3. Action generation structure of a BDI agent [83]

A cognitive agent structured this way, receives external stimuli via sensors and incorporates that as knowledge into its beliefs; after which, it is used in combination with agent's desires to generate its intentions. This type of architecture is simple because there is only single point of control, independently of the situation, the decision-making process considers the input in combination with current knowledge and generates and output action. However, this approach of a unique reasoning unit that knows how to solve and react to every possible situation in the environment is not very realistic and might result in bottlenecks and high response times. Parallel approaches, with units specialized in different levels or types of situations, can simultaneously reason and produce better and faster results.

In this sense, other architectures are hierarchical or layered [83], typically these architectures have two layers and each one of them represents sub-systems that model the agent's behavior, usually its reactive and proactive actuation. These layers can be horizontally or vertically ordered, and there is a flow that defines their interaction (Figure 3.4).

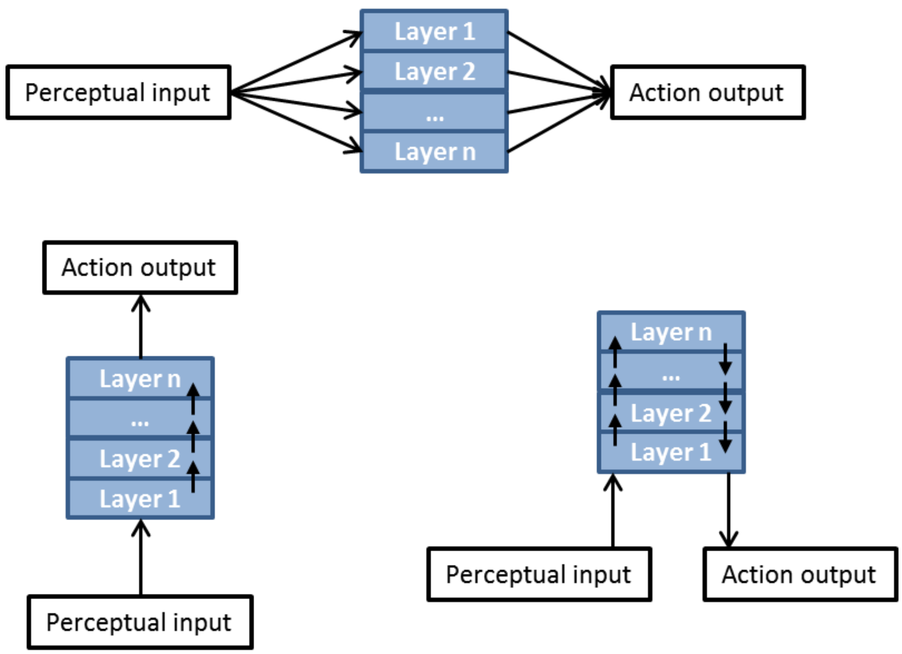

Figure 3.4 Possible agent's architecture layer interactions. Horizontal (top) and vertical (bottom) [83]

In the horizontal case, each layer receives the environment information and works as an independent agent; in parallel, each layer generates a possible action to be executed. In the vertical case, each layer has its own responsibility and its output becomes the input of the next layer; with double control, the output of the last layer is returned through the bottom layers to get an action. With a simple control, the output of the last layer determines the action to be performed by the agent. Even though in a horizontal architecture there is a parallelization of process, this means that a single situation is analyzed several times, once for every layer in the agent; this might lead to a needless over use of the computing power, because the output of only one layer is going to be used as the action of the agent. Such organization requires a pre-processing stage to decide which layer is better suited to solve the situation, or a post-processing stage to decide which of the produced outputs is going to be used. In a vertical arrangement, the first output produced by one of the layers is used and other layers are not called; which can produce better response times, compared to a horizontal configuration.

Implementations of these configurations are Touringmachines (TM) and Interrap (IR), they represent examples of horizontal and vertical architectures, respectively (Figures 3.5 and 3.6) [83].

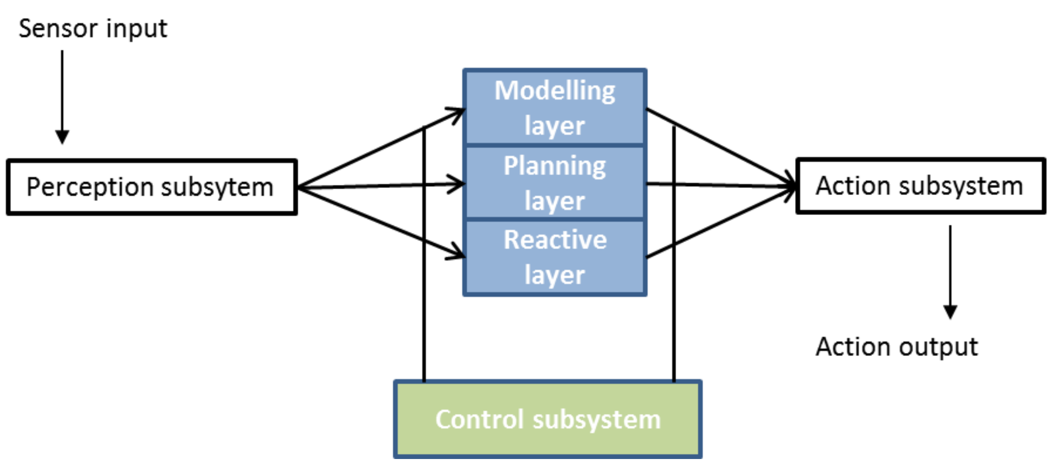

Figure 3.5 Horizontal architecture of Touringmachines [83]

In TM's horizontal architecture, each layer has different responsibilities and generates its own output actions; specifically, the reactive layer is in charge of giving response to changes in the environment (like obstacle avoidance). The planning layer is responsible of the agent's proactive behavior and allows the agent to decide what to do over the base of a previously defined plan library. The modeling layer represents entities that exist in the environment, and it is able to detect situations that can occur on it; it also generates an action plan that will be used later by the planning layer. Finally, the three layers are controlled by the control sub-system who determines which layer has the control of the agent at any given moment.

The cognition of a TM agent distributes different responsibilities among the layers, this enables it, in front of each presented situation, to compute actions at multiple levels; it is also possible to associate priorities to these actions in order to perform the one which is ideal for the agent. Nevertheless, waiting for responses of all layers may increase the agent's response time, which is not desirable in front of situations that require immediate attention.

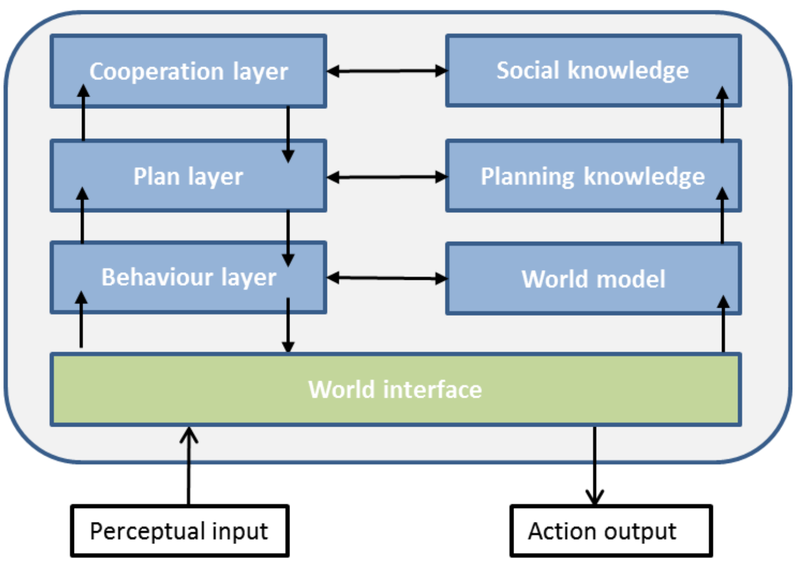

Figure 3.6 Vertical double control architecture of Interrap [83]

In IR's vertical architecture, the bottom layer (behavior) is responsible of the agent's reactive behavior; the second layer (plan) is in charge of achieving the agent's objectives; and the top layer (cooperation) deals with the "social" interactions. Differently than TM, each of the IR layers has a knowledge base which is an understandable representation of the world. So, the top knowledge base keeps information about plans and actions of other agents. The intermediate one has the plans and actions of this agent. And the bottom one holds brute information about the environment. The interaction of these layers begins by the bottom layer; it is activated when it receives direct input from sensors. If this layer cannot deal with the current situation, then the control is passed to the next layer (bottom-up activation). Later, the top layers use the abilities of bottom layers in order to achieve their objectives (top-down execution).

The cognition scheme of Interrap [83] is mainly controlled by the World Interface (WI), it is in charge of receiving new knowledge from the environment, via the perceptual input, and sending it to the behavior layer; which is the entry point of the agent's cognition. Also, it is used to update the context of the agent at each world representation level. Similarly, the WI activates the actuators by generating an action output

This scheme gives higher priority to the reactivity of the agent; the priority is maintained thanks to the layer activation sequence: first the behavior layer, then the plan layer, and finally the cooperation layer. So, the perception and the consequent agent's input are performed in the behavior layer; this configuration is advantageous because the response time will be minimal when the agent has to react to unforeseen events which require immediate actions. Though the use of several knowledge bases can benefit the layer's individual decision making, this redundancy could represent an overcharge in the system.

Another possibility is Brooks [84] proposal in which the control system is decomposed into behaviors in a horizontal architecture (Figure 3.7). All behaviors in this arrangement are connected to all sensors, but each one reacts differently; for any given input, all layers of behaviors are executed in parallel but only one of them will activate the actuators. To decide which action is executed, the author posed a *subsumption* concept in which higher layers have higher priority and subsume the lower layers, by inhibiting inputs or by suppressing their outputs.

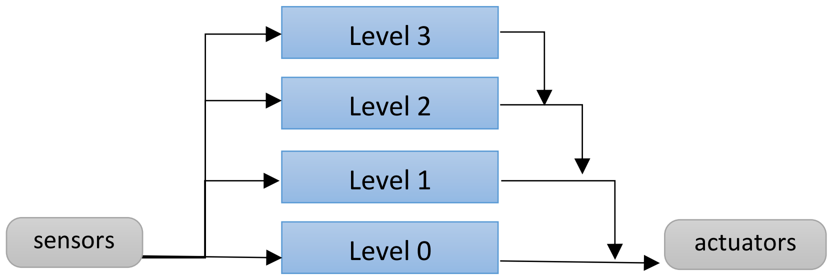

Figure 3.7 Brooks' *subsumption* architecture [84]

Example of control behaviors that the author used for a mobile robot are: avoid obstacles, follow wall, explore and build map; in this particular arrangement, avoiding obstacles has higher priority than building the map. At this respect, Ögren [85] explains two trends: planning vs. reacting; in the first case, also known as deliberative, the agent behavior depends on the world model, has a high response time and can have a high level of AI. On the other hand, a reactive behavior is world model free, has a real-time response and low level of AI.

As in all other presented architectures, the cognition here is fed by the sensors and it acts upon its environment via the actuators; as main difference, the cognition is expressed in terms of parallel behaviors. Being a horizontal approach, this proposal is similar to that of TM; though, this one removes the control subsystem instance and relies on a priority component to decide which of all the possible reactions will be executed. Thinking on the requirements of a model for a vehicle with an autonomous collision avoidance system, there is a maximum delay to obtain an answer from it; after it, maybe the collision cannot be avoided. Thus, removing extra steps in the reasoning process helps to reduce the response time of the whole system, which translates in faster action execution times.

3.2.2.3 Vehicles modeled as intelligent agents

Based on the work of [79], Patron et al. [75] developed a framework using a semantic world model for a hierarchical representation of knowledge; it is used to provide autonomous understanding of the environment to an unmanned underwater vehicle. The semantic knowledge base provides information about the situation, the mission of the unmanned vehicle and its capabilities; furthermore, the rules are used to reason and make decisions before the mission starts, while planning the mission and while executing it. Main contribution of this work is how the knowledge of the environment is stored as a semantic model, which allows the agent to continuously interpret it and decide; it is important to notice also the reasoning process, since it is directly connected to the semantics of the model, which allows it to consider several connections in the data to find knowledge that otherwise it would need the explicit definition of steps in an iterative algorithm.

Zhao et al. [86] present a decision-making process based on ontologies to guide autonomous vehicles to drive safely on uncontrolled intersections. The knowledge base of this approach has an ontology for the map and other one for the right-of-way rules, such as: *stop*, *go*, *turnLeft*, *turnRight*, etc. Instances of the classes are expressed in terms of triples (subject, predicate and object), and rules are described in Semantic Web Rule Language (SWRL). Finally, a rule reasoner is used to infer the action to be taken by the vehicle given the current status of the environment. The research is limited to make decisions like *Stop*, *ToLeft* and *GiveWay*; nevertheless, its ontology description for traffic and rules is an example of how inference logic can be used as the core of the cognition of an agent controlling a vehicle.

The implementation of the COSY modular architecture for agent-oriented traffic simulations, performed by Burmeister et al. [73], is a non-hierarchical approach suited for research purposes concerned by the evolution of the system. With this scheme, modules of the architecture can be easily modified or updated without any harm to other system components. A cognition module in this proposal is internally based on the BDI architecture to rule the agent's decision-making process. It relates its perception to its reactions based on the BDI approach; an agent in this proposal holds beliefs about itself (location, speed, acceleration, etc.), about other agents and about the environment (road length, road width, etc.). Thus, the Cognition module evaluates the data provided by sensors and chooses an appropriate reaction (a plan) when an event of interest occurs). Plans to react to every expected event are previously stored and are queried when the corresponding situation happens; each plan checks conditions in order to decide one action or another, e.g.: overtake, change acceleration. The number of events to which the system can react is limited to three (free driving, following and closing in to another vehicle); also, the centralized element is not desirable if the system is expected to consider several possible situations and several possible reactions to each one of them.

Albus [87] proposes a hierarchical architecture to represent vehicles' behavior; particularly for autonomous vehicles. This architecture is composed of computing nodes, at each level, that contain elements for sensory processing, environment modeling, value judgment and behavior generation. Four levels of reasoning are used to model a single vehicle, at the bottom of this hierarchy (Figure 3.8), information has high resolution, also time and space frames are short; and the opposite at the higher levels. At the top level, the agent plans for the next 50 seconds, next lower level plans for the next 5 seconds; and so on until the lowest level, which has direct contact with sensors and actuators, plans for a 0.05 seconds' horizon. At top of the architecture, the vehicle behavior is also divided into four subsystems: Attention, Communication, Mission package and Locomotion. To model longer term behavior, three more levels are on top of this structure; they allow the vehicle to plan for horizons of 10 minutes, 2 hours and 24 hours, which usually involve communication with similar agents in the same environment.

The author of [87] describes this architecture as hybrid because the hierarchic planning and the autonomous capacity to respond to unforeseen situations, provide a deliberative and a reactive behavior, respectively.

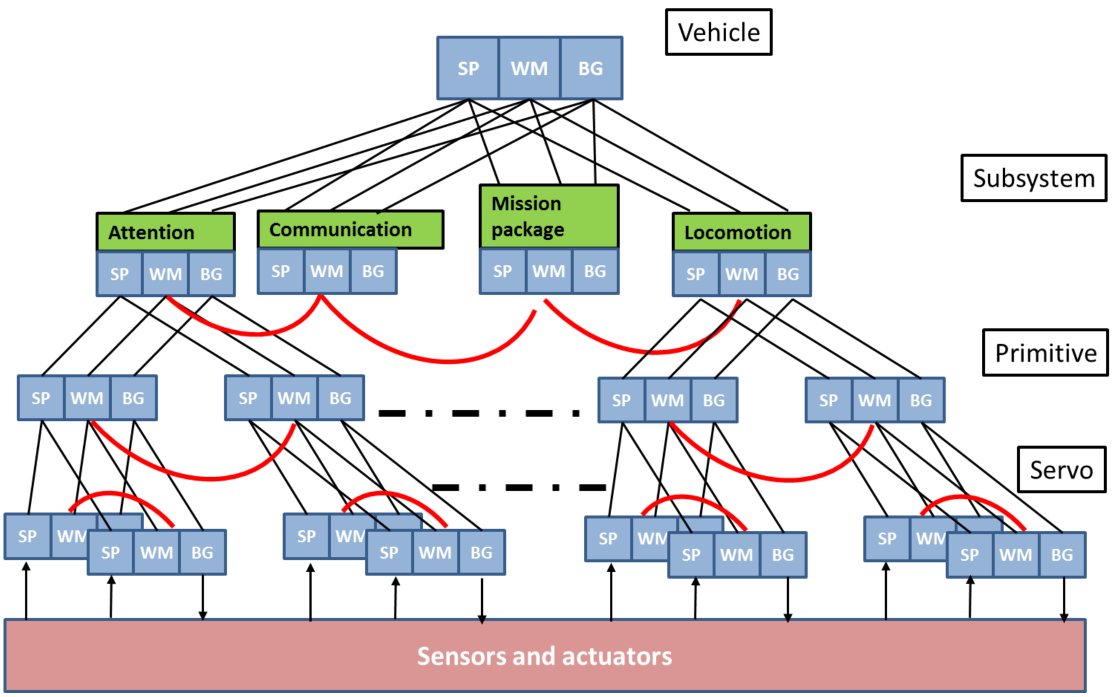

Figure 3.8 Vehicle multilevel architecture [87]

This architecture is very exhaustive by distributing the vehicle's cognitive process into small pieces of responsibilities and layers. Information dissemination is done at an inter-subsystem level (each layer) and at an inter-layer level; this ensures that all components have the most up-to-date information. Even though it is very detailed on what each level is responsible of, as well as the subsystems at each level, the excess of communication and coordination required to act might produce a slow decision-making process; which is not precisely compatible with the response time needed to avoid a collision.

A different approach is presented by Fiosins et al. [88] to model the decision-making process of an autonomous vehicle; the cognition is built in two stages: one strategic and one tactical. The strategic stage is divided in two tasks: pre-planning and routing; in pre-planning, they compute the travel time between all nodes in a graph representing the environment; while the routing task finds the best route to guide the vehicle from its origin to its destination. In the second stage (tactic), authors use speed, lane, traffic light state and distance to the intersection to compute the maneuvers needed to drive the vehicle on the road up to the next intersection; these movements are defined in terms of the variation (Δ) required in current speed and lane $\langle \Delta v, \Delta l \rangle$, each time the vehicle gives a "step" on the road. Such approach, although simple, considers a deterministic environment in which the possible changes of variables are pre-set, like the traffic lights or the vehicle's speed; it does not consider the presence of other vehicles or the occurrence of unforeseen events.

Also, the work of Wahle et al. [89] has a scheme of two layers, in this case to model the decision-making process of a driver agent (Figure 3.9). The reactive layer describes perception and reaction of the agent within a short time scale; whilst the strategic layer is responsible for information assimilation and driver decision making, e.g.: select a route from several alternatives.

Even if the last two presented works do not detail on the reasoning process, it is important to notice the simplicity of the cognition modules and the concrete distinction between architecture layers; using the time as basis, those tasks with a long-term horizon are on a strategic level. Other tasks, requiring immediate action are on the reactive level. Thus, path or route planning are on the higher level, while driving and collision avoidance are on the lower level.

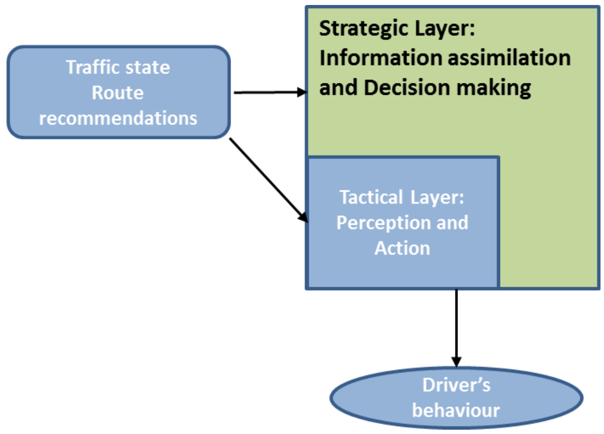

Figure 3.9 Two-layer scheme presented to model a driver agent [89]

Given that vehicles are not alone on the streets, achieving a cooperative behavior is a necessity. Thus, it remains pending how communication capabilities are to be integrated to the model; this is key to reach a cooperative behavior, in which intentions and objectives of several vehicles must be taken into consideration.

3.3 Modeling of cooperative capacities

Applications using intelligent agent technologies for traffic and transport systems are divided in five categories [90]: traffic management and control architecture, agent systems for terrestrial transportation, agent systems for aerial transportation, agent systems for rail transportation, and multi-agent traffic modeling and simulation. Chen & Cheng [90] mention the works of Garcia-Serrano et al. [91], Tomás & García [92] and Chen et al. [93] as Multi-agent systems (MAS) contributions for traffic congestion detection and management in compliance with the IEEE Foundation for Intelligent Physical Agents (FIPA) standards; these works are particularly interesting given their contribution to the platform definition and system architecture. In this first category, systems are classified as hierarchical, *heterarchical* or hybrid; the former divides the system in small sub-systems with interaction between them; distributed approaches, with agents communicating between them to perform decision making, are referred to as *heterarchical*; and, hybrid systems combine characteristics of the other two types.

Regarding the second category defined in [90], agent systems for terrestrial transportation are usually classified as centralized or decentralized. The first ones must compute and assign all tasks in the system to available agents in the network; this has the consequent drawback of complexity enlargement when the number of agents increases. The decentralized approach proposes that mobile agents in the network make independent decisions to create distributed control systems. Finally, authors mention that other terrestrial transportation control systems use several types of agents representing different real-life entities and their interactions, for instance: traffic lights, traffic signals, street lanes, taxis, bus stops, etc. Most multi-agent system approaches search the best route for vehicles according to current known traffic conditions, e.g.: trying to avoid zones with high vehicle density; furthermore, if a congestion situation already exists, the system can generate solutions to exit from it. As it happens with every centralized solution, it is limited by the maximal number of clients it can process at any given moment, so there could be bottlenecks and therefore delays when generating plans for the vehicles. Conversely, making decentralized decisions, using only local information, allows better response times at the cost of finding suboptimal solutions.

Agents representing infrastructure have information related to the use of the road segment they are on; they can transmit this information when it is demanded. Agents can have the exploration or intention responsibility; each vehicle uses several exploration agents, in the network, arranged as an ant colony. This organization conforms a communication link with agents requesting street status and delay values, as well as other agents updating them.

On this topic, Düring and Pascheka [94] claim literature is ambiguous on giving a single definition for cooperative behavior. According to them, some authors agree that agents with cooperative behavior must have the ability to work together; which requires them to have communication components for exchanging, or at least for receiving, information. However, inter-agent communication can be implicit; thus, it is not necessary to have explicit communications capabilities. Similarly, there is no consensus in the objectives of cooperative behavior, it could exist a common good or a common goal. To clarify in this matter, they propose seven properties to define cooperative behavior:

- "Cooperative" is an attribute to an agent's behavior.
- Requires the existence of at least two agents.
- Requires a concept of utility.
- Affects another agent's utility.
- Affects the acting agent's utility.
- Requires knowledge and will.
- It is relative.

Key concept to consider here is the *utility*, authors explain that cooperative behavior has a positive connotation; so, for a behavior to be cooperative it has to have some usefulness for the agents. Therefore, they must have a utility function. Given this utility, an agent's behavior can be driven to increase its own or that of other agents, i.e. egoistic or altruistic, respectively. Moreover, if the behavior, knowingly and willingly, increases the total utility of the system it is considered as cooperative, or uncooperative otherwise. Finally, if the behavior leads to increase utilities of both agents it is denoted as rational.

So, the utility notion proposed in [94] is useful to stablish the behavior of agents in a cooperative environment; providing specific directions to define agents as cooperatives or not, in order to increase the utility of the whole system, is a significant contribution; specially for the type of situations that the present research intends to avoid, in which the final goal is that none of the involved agents gets damaged.

3.3.1 Communication

One of the main characteristics of cooperative driving is the use of combined information from several vehicles, which could increase perception capacities of each node of the network. This collaboration can also serve for entertainment inside the vehicle, to know traffic status in distant areas where the driver is heading to, or even to detect dangerous situations in the immediate environment. Wang et al. [95] perform cooperative localization based on connected vehicles capacities. The goal is to use joint information, so each vehicle is able to create a very accurate map of vehicles around it. Each car in this scheme is equipped with a 2D laser scanner, a GPS, a digital compass, a stereo camera and a DSRC communication unit. They use a broadcast approach to send messages with location and motion information among cars; later, each car integrates the received information with its own, using an algorithm for simultaneous localization mapping and moving object tracking (SLAMMOT). It represents detected stationary and mobile objects as data points in a map; more details on this algorithm can be found in [96]. An important contribution of this research is the introduction of a cooperative approach for precise localization, based on the fusion of self-collected data and that coming from others in a single obstacle map. As an additional feature, authors implemented a face following system with a camera inside the vehicle to detect when the driver is distracted, and inform to neighboring vehicles; which is an interesting contribution for minimizing danger on the roads by warning other drivers so they can prevent a collision, it is also an example of the kind of the attainable benefits with communication capabilities in vehicles.

An intelligent system may be composed by one or several agents. When more than one agent is needed to execute a task, we talk about a multi-agent environment; in this case, besides the internal conception of the agent, it is necessary to define its inter-agent communication and coordination. As it will be detailed later in this chapter, one of the first steps is to decide if the generation and allocation of sub-objectives will be centralized or distributed. The next characteristic to consider is the interaction between the agents; are they going to compete each other to accomplish independent tasks, or will they cooperate in order to complete a common global goal?

According to Huhns and Stephens [81] a multi-agent environment is characterized by the incorporation of protocols for the communication and protocols for the interaction; the first ones allow sending, receiving and understanding messages, and the second allow the agents to exchange messages in a structured manner. These protocols finally allow the coordination among the agents, which can be to cooperate or to compete, as can be seen in Figure 3.10.

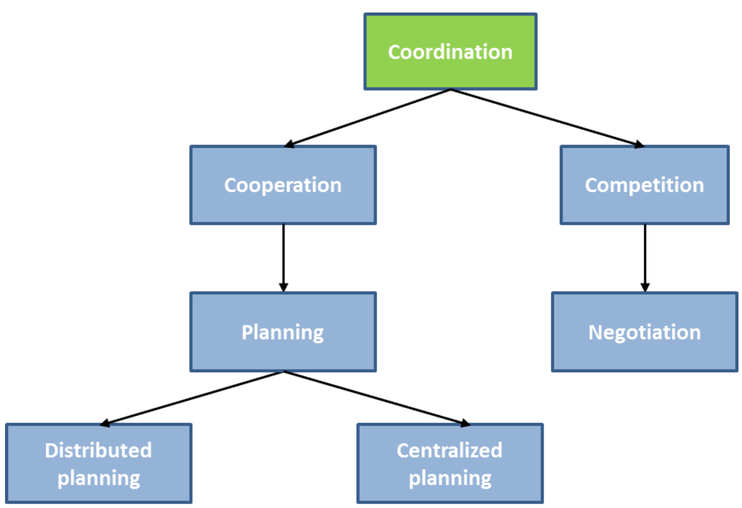

Figure 3.10 Coordination of agents' behavior [81]

In a multi-agent environment, where the agents have similar objectives and where the decision-making process must be kept distributed, it is important to define the shared objectives and the common tasks, to avoid conflicts and to save the knowledge and the collected evidence [81]. Authors of [81] describe the contract net and the blackboard system as two approaches conceived to find the best allocation of tasks to agents in the network; the former is based on communication between the agents and the latter is based on a shared data structure. In the case of the contract net, there is a manager agent in charge of announcing tasks and receiving offers from contractor agents, it will decide which is the best agent to perform each task; while in the blackboard system, tasks are stored in a data structure, accessible by all, and are the same agents who decide which task to execute, given their own abilities. As explained before in this section, such centralized way to assign tasks represents a common concurrency point which could be overcharged when the number of tasks or agents increases, and represents a weakness of both methods; however, the blackboard system reduces the dependency on the centralized component by leaving the agents independently decide what to do.

As it was previously exposed, goal accomplishment in a multi-agent system is based on agents' interaction; it can be direct like in a contract net or indirect like in a blackboard system. Dimopoulos and Moraitis [97] propose two algorithms, one for agents' coordination and another for cooperation. According to their explanation, coordination implies that individual agents have different goals to achieve; also, they have to be able to perform them on their own. Furthermore, in the case of conflicting plans, agents have to find a way to modify them so they achieve their goals without impeding those of other agents. On the other hand, cooperation implies that agents might not be able to perform tasks on their own and that they have to ask other agents for help in order to accomplish them. In both cases, it is possible to have agents with different abilities. This discrimination by types, according to the interaction with others, is essential to determine how agents in the system deal in order to accomplish their individual tasks; likewise, it results useful to stablish a level of cooperation or collaboration depending on the situation the agent is facing at a given moment.

Plans generated by [97] are coded in propositional logic according to the SATPLAN (Kauts et al. [98]), which is an algorithm for plan generation that uses: the initial state of an agent, the set of agent's goals, the maximal plan's length allowed limit and the set of constraints. The multi-agent coordination proposal of Dimopoulos and Moraitis [97] indicates that each agent computes its own action plan (by calling SATPLAN); then, this plan is sent to other agents to be used as candidate sub-plan for a new global plan. The global plan is supposed to incorporate the goal of two agents. To compute a new global plan, each agent (A) receiving a plan of another agent (B) realizes a new call to SATPLAN, but with different parameters: the initial state is made up of the set of states of both agents plus the possible interactions between their current plans, and the constraints include those related to the second agent's (B) plan.

There is an advantage in terms of the response time when using propositional logic to code the plans; also, it is certain that if the constraints are well defined, the solution for any given plans intersection will be valid because of the logic structure.

Coding a configuration of a real-life street situation in terms of propositional logic is not an evident task, because it has to include several elements that interact with the vehicle and with each other, such as: the street, other vehicles, pedestrians, intersection, traffic lights, other obstacles, etc.; the complexity of writing all possible relations between the elements would increase as the number of elements does. Moreover, in a collision avoidance scenario, missing or not considering an important interaction could have major consequences for the lives of the drivers and pedestrians. Another drawback of having the plans in such format is the necessity of an extra process to translate them to concrete vehicle actions.

So, communication capabilities can be considered as a high-level feature; they are mainly used to obtain knowledge of the environment that is not reachable by ego sensors. However, after sharing information, an agent must fusion incoming data with its own in order to keep one single image of the reality; by integrating other agents' intentions with its own, it can detect conflicting plans. Therefore, to solve such type of problems, further interaction will be needed; one that implies a plan synchronization process.

3.3.2 Synchronization of agents' plans

Intelligent agents' ability to communicate in a system allows them to share information that otherwise they could not know; however, the job is not done yet, after its reception, the agent must verify if it is useful for the completion of its individual objective, this might also involve to fusion data coming from multiple sources in order to reduce redundancy and the complexity of the analysis. Thereafter, the agent is able to generate an action plan according to its objectives; as was exposed before, an input for this generation is the information provided by ego sensors and from neighbors. Therefore, bearing in mind the definitions presented in previous sections, an agent that synchronizes its actions with others in a shared environment can be considered as cognitive; furthermore, as the works exposed in this section study agents with capabilities for perceiving their surroundings, integrate that new knowledge to their beliefs, and use it to act over it again, we can consider them as cognitive approaches for cooperative plan synchronization.

Since an agent shares the environment with others, its action plans have two main constraints:

- Not interfering with the global objective of the system (if it exists), and
- Not to be in conflict with the plans of its neighbors.

Thus, by considering particular and general interests, agents involved in a common situation must come to an agreement, or “consensus” [99] with their respective plans. Olfati-Saber et al. [99] describe a consensus algorithm as the set of rules for the exchange of information between the agents and their environment.

Both centralized and distributed approaches are found in the literature to synchronize plans in a multi-agent environment. The choice between one or the other considers the type of task to be assigned to the agents, which directly relates to the maximal timeframe size to generate an action plan. In his work, Albus [87] explains that goal-seeking tasks are related to a reactive behavior with short intervals of time and space; for medium and long-term tasks, the range of time and space increases, and therefore it can be handled at higher levels and computed less frequently. Therefore, as route planning and traffic avoidance capabilities allow longer response times, they can be performed in a centralized manner; such approach benefits from updated information coming from multiple zones in order to organize a high number of vehicles on a city level. On the other side, reactive tasks, such as collision avoidance, planned in short timeframes use only local information and are usually performed in a distributed way at a street level. The synchronization of plans of different agents occurs only at a high level, when agents in the system are planning their assigned tasks; author of [87] exposes that this synchronization involves communication between components of the same level of cognition of several agents, where they select goals and set priorities for them. Though, plans at the reactive level are not synchronized; for the case of conflicting circumstances, like the risk of a collision, agents react independently, using sensor data without considering possible collaborative actions to resolve the situation.

Qu et al. [100] propose a co-evolution strategy to synchronize the trajectories of several robots in a shared environment. Each robot implements a genetic algorithm, with its own population, to compute its own route; the chromosome is composed of the ids of the positions where the robot has to pass on the shared grid. After each iteration, all robots share genetic information of their best individuals in an island model. A second fitness function is used at this stage to evaluate multiple robots' trajectories as a unique global solution; this function assigns higher fitness values to those routes more synchronized with other robots' routes, i.e. routes that have fewer collisions with others. Thus, best individuals of each population are evaluated according to their best adaptation to the global solution. Finally, the best individuals obtained from this synchronization process are directly used as part of the next iteration's population in the corresponding robot. The flow diagram in Figure 3.11 depicts this process.

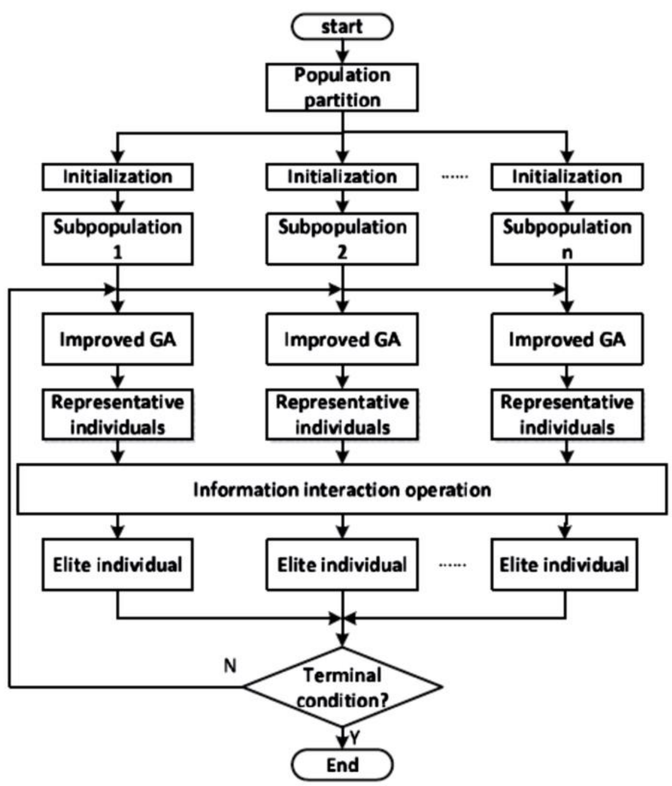

Figure 3.11 Co-evolution algorithm's flow diagram [100]

This work presents an interesting way of centralized synchronization; the use of an evolutionary algorithm has the advantage of fast searching in a set of possible solutions, in this case for route finding, and the combination of genetic material between the different populations guarantees non-overlapping plans. This configuration fits the requirement of completing the particular plan without conflicting with other agents' plans. However, even if evolutionary approaches, such as genetic algorithms, give faster results than an exhaustive search, they might not be fast enough to find the best solution for a collision avoidance situation, which requires a reaction within seconds. Also, even if there is an intrinsic cognitive component because of the exchange and integration of information between agents, it is not explicit nor independent; the synchronization achieved by the fusion of data of all agents, controlled by a central unit, produces a collateral cognitive characteristic in the model.

Claes et al. [101] present a solution to compute the routes of several vehicles of a delivery company; packets may pass by depot centers before arriving their final destination. This solution has a hierarchical structure of three levels; the hierarchy is related to the transport network range. The top-level coordinates aerial transport actions; the second coordinates actions in a region; and the bottom layer coordinates specific actions of delivery trucks. This scheme proposes that each package to be sent (represented by an agent) activates a set of agents on the superior level, which conform an ant colony system in charge of finding the best delivery route for the package, considering time and cost constraints. Agents on the intermediate level obtain information about capacities at each depot center, through agents representing these latter. At this level, two types of ants (agents) are used, one that searches the best route to deliver the packet; and another one that returns to confirm intermediate nodes (depot centers) that the route will be used and that they have to reserve space for the packet. Each node at the second level has information about trucks' departing times; such information serves to determine if a packet can arrive on time using a route connecting trough this center.

As was mentioned before, metaheuristic solutions provide suboptimal solutions that could take a considerable amount of time. In this case, they are used to gather information from different sources and to find the best configuration of routes for several agents. The ant colony system used to find the routes serves, at the same time, as a mean of implicit communication between agents, to synchronize their plans in order to achieve the global goal. An important contribution of this work is the definition of layer dependent populations, each one having its own level of world understanding and objective accomplishment. This particular arrangement permits the creation of local solutions, each one solving a small part of the problem; synchronized later to complete a global objective. Furthermore, the proposal as a whole can be viewed as a system with cognitive levels, each one completing a stage of the problem; later, the integrated results provide a global solution.

3.4 Collision avoidance

This section of the chapter is intended to discuss works on one of the main research axes of this study: collision avoidance. Initially, we explore advances on individual collision avoidance; this is, the design of collision avoidance capacities for a single vehicle. Capacities, that can be later adapted for collaborative environments. Afterwards, a sub-section is dedicated to examine cooperative driving approaches, with special focus on intersection crossing, collision advisory and avoidance.

3.4.1 Individual collision avoidance

In order to be able to avoid a collision, a vehicle must have knowledge of what is happening in its surroundings; sensors of several types can provide the necessary detailed information for that. Jimenez et al. [102] use ultrasonic sensors on the rear to provide information to a collision avoidance system that performs braking and steering maneuvers. As exposed in the previous chapter, this is an example of a connected vehicle, with a laser scanner on the front and a GPS receiver that complete the detection system; this configuration allows them to locate the vehicle, obtain the distance to another vehicle in front, and to compute the relative speed between the host and a vehicle moving on the adjacent lane. Three scenarios were considered to test the feasibility of this arrangement, for all of them the vehicle with sensors was in collision course with another vehicle in front, and with an approaching vehicle on the adjacent lane. Authors created a series of conditions to comply in order decide a braking or an evasive maneuver solution. Using equation (3.3) the system decides if it is safe to apply the brakes in order to avoid the collision.

$$d \geq (v_{1-}v_2)t_{r1}\frac{(v_1 - v_2)^2}{2a_1} \tag{3.3}$$

Where, d is the distance between the two vehicles, v_1 and v_2 are their speeds, t_{r1} is the reaction time between when the vehicle 2 is detected and the braking process starts, and a_1 is the deceleration of vehicle 1. If this equation holds, it means that the speed of the first vehicle can be reduced to that of vehicle 2. Similar equations are used to keep a third vehicle away from collision with vehicle 1 if it changes lanes. Details on obstacle detection and position system of this work using only one frontal sensor can be found in [103]. The proposal is limited to consider only one car having sensory capabilities in a highly deterministic scenario; also, it relies on computed values based on data provided by sensors, communication capabilities are not considered to share speed, location, orientation or other vehicles status data.

Lee et al. [104] explain that the Time To Collision (TTC) is used to determine the risk of a collision. There are various ways to estimate and use the TTC; it is usually computed as the remaining time before a collision takes place if vehicles continue with their current speed. When the TTC is below a specified threshold warning and collision avoidance systems can be triggered. Even if the TTC indicates how much time does a car have to avoid a collision, authors of [104] also indicate that it must be used in combination with the distance; so, cars should respect a minimal distance, also known as *safe distance* (SD). The safe distance between a vehicle (host) and another in front on the same lane (preceding) can be computed according to equation (3.4).

$$D_s = \frac{v^2}{2a} - \frac{(v - v_r)^2}{2(a - a_r)} + D_m \tag{3.4}$$

Where v and a are the speed and acceleration of the host and preceding vehicle, respectively; v_r and a_r are the speed difference and acceleration difference between the host and the preceding vehicle, respectively; and D_m is a minimum acceptable distance when both vehicles stop.

According to Milanés [105], it is generally accepted that a collision risk exists when the TTC is 2 seconds or less; and Lee et al. [104] consider that a collision risk is present when TTC is 1.6 seconds and the distance between vehicles is smaller than the safe distance (3.4).

In this sense, Jimenez et al. [102] propose to perform a steering wheel maneuver if there is not enough time to brake; in this way, a front collision with a car going on the same or the opposite way can be avoided. Also, Ackerman et al. [106] present a solution that considers braking and swerving. The collision detection system is based on information provided by two radar sensors in the back, one radar sensor and a camera in the front; to fuse this data, the system selects relevant objects in the environment and matches them in an integrated outlook. The system was designed to act in the last minute if the driver has not reacted to the collision situation; to achieve this, they have computed the time to collision (TTC) considering that both vehicles continue with their current speeds. Then, with the TTC it is possible to calculate the time to brake (TTB) and time to steer (TTS); which are the time after which a braking maneuver or a steering maneuver must start to avoid the collision, respectively. The decision-making process, considers these times and information about the free space in the other lane in order to select the best avoidance maneuver. The research does not consider the use of a simultaneous combination of braking and steering maneuvers, which might give a broader spectrum of collision situations to avoid; just using the brakes to avoid a collision might not be enough in all cases, specially those where there are vehicles coming behind. Even if some collision situations can be avoided just by applying the brakes, some others may well be more complicated and need a complex maneuver that brakes the car and turns the steering wheel.

A significant contribution of these researches are the proposed formulae to compute the remaining time for the collision to take place, the remaining time for a braking maneuver and for a steering maneuver. These are of main importance for a cooperative collision avoidance environment, since they provide thresholds to be held while performing the stages of evasive maneuvers: communication, negotiation and execution.

In order to safely avoid a collision by executing a lane changing maneuver, knowledge of the vehicle's kinematics model is required, as well as information on the environment involved in the maneuver; a control model must guide the car from its current place to a safe, centered point on the next lane. At this respect, Naranjo et al. [107] propose a lane-change fuzzy control for autonomous vehicles that considers the lateral error and the angular error. Lateral error is the distance of the center of the vehicle to the center of the lane, and angular error is the angle between the vehicle's direction vector and the lane. Their controller outputs a steering direction, left or right; and, authors of [107] have determined that, for safety and comfort reasons, the steering change at each timestep must not exceed 2.5% of the maximum steering angle. Figure 3.12 depicts the membership functions for the linguistic variables Lateral-error and Angular-error proposed by [107]. Fuzzy systems are commonly used as robot motion controllers, and thanks to their immediate response they are suitable for situations where the time response is critical, such as collision scenarios. Even if the work of Naranjo et al. [107] is not intended for collision avoidance, the logic behind their controllers serves as a basis for our research.

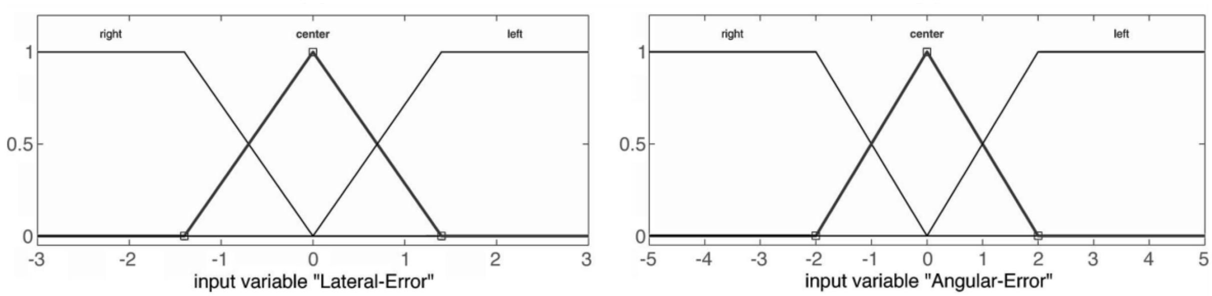

Figure 3.12 Membership functions for Lateral-error and Angular-error [107]

Similarly, Guo et al. [108] designed a controller for lane changing trajectory planning and tracking on a curve road; to incorporate the curve variable while changing lanes, the authors considered a non-holonomic constraint and a tracking error model. The strategy used to compute the lane change trajectory is based on a trapezoidal acceleration profile; which, according to the authors, generates the least possible lateral acceleration of the vehicle. For the matters of tracking control, both world and local coordinate systems were used; and to compute the instantaneous rotation center, they used the kinematics provided by the Ackerman vehicle model. This study is an important contribution to the control mechanisms for autonomous vehicles on curves, most of the researches make efforts on simulating vehicles on straight streets; moreover, the work of [108] provides required formulae to control vehicle's kinematics to make a smooth transition from one lane to another while avoiding to crash with the borders.

3.4.2 Cooperative collision avoidance

Thanks to the introduction of new features into the vehicles, specially those integrating communication capabilities, the researchers are more and more interested in testing them, and their possibilities; this has motivated the study of solutions for cooperative driving. Collision-free intersection crossing represents a largely explored problem on the cooperative driving domain; proposed solutions in the literature use information delivered by incoming vehicles to take decentralized decisions or via a central controller.

An example of this is the work of Lee and Park [57] for intersection management for autonomous vehicles; a control module at an intersection calculates the best traversing maneuver by predicting the distance that has to be respected between each pair of vehicles while crossing. This distance is computed based on acceleration values. The computation of each vehicle maneuver is considered as an optimization problem which objective is to avoid trajectories' superposition while they are over the intersection. This approach is based on two analytic algorithms (ASM and IPM) and an evolutionary algorithm (a Genetic Algorithm); these algorithms run sequentially to find a feasible solution so that all vehicles traverse the intersection collision-free, the first solution found is applied. Lee et al. [109] have extended the algorithm in [57] to use it in simulations of a corridor with several intersections; based on the same premises, results of this implementation show a considerable reduction in values related to delay times, CO2 emissions and energy consumption. However, there are a number of limitations of this approach; first, the necessity of a central coordinator at the intersection that computes and indicates the maneuvers of each vehicle. Secondly, the sequential execution of the solving algorithms is an important drawback because vehicles are continuously approaching to intersections, and waiting for a solution can lead to traffic jams; moreover, there is not guarantee that any of the algorithms finds a feasible solution in the expected time of arrival of the vehicles to the intersection. Finally, authors of [57] assume 100% of automated vehicles with communication capabilities within their scenarios; which will not be realistic because the introduction of this technology is occurring in a progressive manner.

Abbas-Turki et al. [110] propose an algorithm to avoid collisions at an intersection crossing, by controlling the right of way considering access priorities. In this particular project, the objective is to optimize public transportation schedule times. The system works as follows: first it refuses the pass to all nodes, organizes them in separated queues and assigns their right of way according to priority. All possible valid passing assignations are organized in trees, so the algorithm can assign the pass to several nodes at the same time (those without conflict). This work is limited to use only expected arrival times of public transportation units in order to assign the priorities; also, the proposal does not compute evasive maneuvers or velocity changes to vehicles approaching the intersection, instead of detecting the danger of collision and avoiding it, this approach prevents the occurrence of collision situations.

A similar approach is the work of Wu et al. [56] for Autonomous Intersection Management (AIM). According to the authors, and as it is expected, this system has to give results in real time; so, brute force strategies are not a suitable option. Therefore, their solution recovers information from nodes via V2V and V2I communication and later an Ant Colony System algorithm (ACS) is used to determine the pass order of the vehicles, considering arrival orders. Again, this proposal avoids the danger of a collision by using metaheuristics to find a suboptimal solution for a scheduling problem; so, it does not offer avoidance plans to prevent a collision, but provides the instructions to the vehicles to cross the intersection collision-free. Such approaches like [110] and [56] are not taking full advantage of connected vehicles capabilities, since they are just a highly specialized traffic light that decides who can pass and who has to stop.

Ahmane et al. [111] use three sensors on the roads to detect the occurrence of events of interest in an intersection: an incoming vehicle to the intersection line, the arrival of a vehicle to the intersection and the exit of an intersection. Vehicles are modeled according to a Petri network; they consider a set of simple rules to assign the pass order, among which: minimal time space between vehicles and the consideration of two simultaneous flows of traffic (west-east and north-south).

An important contribution of this work is the distributed approach to assign the “right of way” to vehicles; basically, they are allowed to cross the intersection in the same order they arrive to it, which is captured by sensors on the road. Another interesting idea is the creation of clusters of vehicles that are near each other and do not have conflicting flows; this allows the system to simultaneously assign the “right of way” to them in order to improve flow efficiency in the intersection. However, there is a scalability issue of this proposal because they require three sensors installed at each street approaching the intersection; doing so at each intersection would require the expense of economic resources to improve a city’s infrastructure.

As many crashes occur at intersections, both with traffic lights and stop signs, another way to minimize their occurrence is to detect if an approaching vehicle might get involved in a collision.

Maile et al. [112] explain in their work that, using V2I communication, an equipped infrastructure can compute if an incoming vehicle will not stop at the intersection or even if its current speed will not allow it to stop on time before the light turns red. If such a situation occurs, the equipped traffic light can send messages informing this to the vehicle and to other incoming vehicles, so they can take proper actions to avoid a crash. Even though the proposal is not designed for autonomous vehicles, it considers two important characteristics for an early adoption of the technology: vehicles can be informed via midway hops, the infrastructure in this case; and secondly, the dissemination of information about potential collisions according to the other vehicle's status. However, the study does not consider heterogeneous scenarios, those involving both connected and non-connected vehicles.

Maile et al. [61] propose a prototype of Intersection Collision Avoidance System (ICA) based on DSRC communication; the vehicle with collision avoidance capabilities, called Host Vehicle (HV), receives information from surrounding vehicles through an embedded platform: The Wireless Safety Unit (WSU). The system can act over several configurations of intersection scenarios, such as: straight crossing, right or left turn into path, lateral direction or opposite direction; which is very innovative, because they are different from the most commonly considered scenarios in collision avoidance researches. When approaching to an intersection, the HV receives broadcast messages from other incoming vehicles, and uses it in combination with other on-board systems; these other systems include a GPS, a heading sensor, vehicle dynamics sensors for speed, acceleration and yaw, vehicle status sensors for brake, transmission and turn signal. All of them integrated in a single software unit (called Wireless Safety Unit, WSU) that keeps position, path history and predicts path to compute the possibility of a collision.

The ICA application is in charge of fusing the remote and local data to perform a situation analysis; if a collision scenario is detected, the threat evaluation module determines the threat level, which can be high or critical, depending on distance and needed deceleration thresholds. For high level threat, the system sends acoustic and visual warnings to the driver; if the driver does not react to the signals, the threat level increases and the system orders the vehicle maneuver control to apply the brakes. The execution of the brake maneuver is performed by the vehicle control unit which is an interface with the car's electronic brake system. A flow diagram of this system is presented in Figure 3.13.

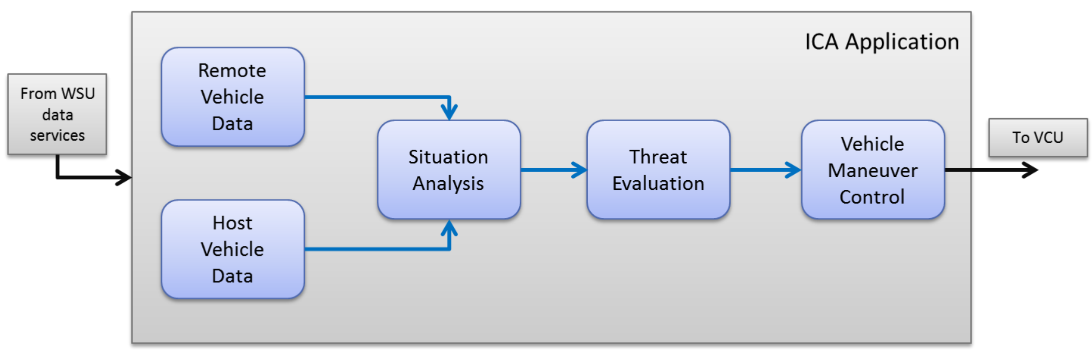

Figure 3.13 Details of the ICA Application [61]

Each scenario is treated in a different way, and for that, the authors have defined a set of algorithms that asses the type of situation. The ICA Application in this research only can resolve to apply the brakes as a solution for a collision situation; also, in exposed experiments the proposal does not explore the activation of collision avoidance capabilities on both vehicles present in the scenarios.

Another solution for collision-free cooperative driving is proposed by Caveney and Dunbar [113]: The Distributed Receding Horizon Control (DRHC), it performs a shared decision making using V2V communication. To illustrate the use of technologies allowing it, they test two applications of autonomous driving: platooning and cooperative merging. Their framework integrates a group coordination logic based on events, periodic route planning, digital maps, collision avoidance and communications. Each vehicle in a distributed manner uses its own status information to predict the trajectory for the next five to ten seconds in the future; this “assumed” trajectory is then shared with vehicles in the proximity via V2V. The collision avoidance process is based on a set of logic conditions to determine how the expected trajectory of the vehicle is going to overlap those of vehicles around; which is one of the major contributions of the proposal. Nevertheless, a standardized environment is assumed, where all vehicles are supposed to have communication capabilities. Also, the vehicle model used to compute trajectories and avoid collisions is over simplified, as it considers only the (x, y) coordinates and the heading angle; such simplification of variables limits the reach of this research because it does not use other kinematic constraints of the vehicle, or the status of the road and other elements of the environment, to produce realistic results in the computation of the future location of the vehicle.

Sun et al. [114] propose a collision avoidance approach, similar to the behavior-based of Brooks [84], but specifically designed for a multi-robot environment. In this research, behaviors are defined for allowing a robot to follow a waypoint, avoid another robot, pass first over a trajectory intersection point, wait for other robot to pass over an intersection point and for keeping distance. Each one of the behaviors has associated an algorithm to rule it. Authors define a frontal area and a critical area for the robot, the first one is oriented towards the desired robot direction (the next waypoint), and the latter is the drive channel needed by the robot. Robots in the environment continuously exchange location, path and status information with their neighbors. Moreover, they use a function to continuously check if other robots are already situated in its way; in which case, they can decide to trigger one of the mentioned behaviors. Main contribution of this research is the definition of a set of behaviors, independently activated depending on the situation recognized by the robot; so, instead of activating them all at once and later decide which action to execute depending on a behavior priority, the robot uses the knowledge it has on its environment to decide which behavior is most suited to current circumstances.

The cooperative collision warning system of Huang and Tan [115] is based on a future trajectory calculation of vehicles involved in a collision situation. Their system estimates and communicates vehicles' locations; with this information, it is able to compute future trajectories of the vehicles and determine the possibility of an intersection in space and time. Vehicle localization is based on a DGPS unit; the communication was simulated as using the bandwidth 5-20Hz, and protocols or devices were not specified. The predicted trajectory is given by a set of states, in which the vehicle's future state is predicted through model-based propagation; using current location and the look-ahead time. To compute a potential trajectory conflict, the system uses predicted trajectories to compare the future distances between the host and the surrounding vehicles within 2-3 seconds in the future. They defined D_{p12} as the minimal distance between any two cars, and the probability of a position conflict as:

$$Pr_{pc} = prob\left(D_{p12} \leq D_{th}\right) \tag{3.5}$$

Where D_{th} is a pre-defined minimal threshold after which a trajectory conflict might occur. And the probability of a position conflict in the future is:

$$Pr_{pc}(t_n, t_k) = prob\left(D_{p12}(t_n, t_k) \leq D_{th}\right) \quad (3.6)$$

Where t_n and t_k are, respectively, the current time and the look-ahead time in prediction. Thus, a potential trajectory conflict is triggered if the Pr_{pc} for any two cars is higher than a pre-defined Pr_{th} threshold. And the corresponding time-to-position-conflict is:

$$TTPC = \min t_k \ \ , t_k \in \left\{t_k | Pr_{pc}\left(t_{n,} t_k\right) \geq Pr_{th}\right\} \quad (3.7)$$

The probability and persistency of potential trajectory conflicts can derive in the detection of a potential collision. According to the authors, the persistency of a trajectory conflict is defined as the time that its detection persists. So, if a conflict detection persists over a threshold $T_{persist}$ and the trajectory conflict has reduced the time-to-position-conflict (TTPC), then a potential collision warning is triggered. The authors include in their calculations a prediction error, which refers to the difference between the predicted trajectory and the real one; they explain that it can be produced by errors in the assumptions of the driver's input to the system.

A comparison of the efficiency of a cooperative driving advisory system and a cooperative driving autonomous system was made by Broek et al. [62]. This work considers acceleration variables, the inter-vehicular space and the communication delay. According to their conclusions, autonomous vehicles generate an increment in vehicular flow by reducing headway time and keeping string stability and speed. Also, a collision warning system is presented by Sengupta et al. [50]; it is based on wireless data to recognize the environment of each vehicle. Instead of using sensors, their system considers current data about speed and orientation exclusively using communication capabilities; vehicles must share, with their neighbors, their own location information and also infrastructure-related information, such as state of traffic lights, permitted turns, etc.

Fujimori et al. [116] define five unique situations where a collision will occur if current state continues, i.e. if none of the robots involved acts to avoid the collision. These situations are depicted in Figure 3.14.

Where, R_1 and R_2 are the robots, v_1 and v_2 are their respective velocities, and δ is the crossing angle. Authors propose a velocity and a direction control to modify current state of robots and avoid the collision; however, for the case of Figure 3.14(a) only a change of direction can effectively prevent the collision from happening. They include a priority component to decide which vehicle will pass first by the trajectory intersection point; each of the robots has as priority value the inverse of their arrival time at the crossing point:

$$w_i = \frac{v_i}{\overline{R_\iota P_{12}}}, (i = 1,2) \tag{3.8}$$

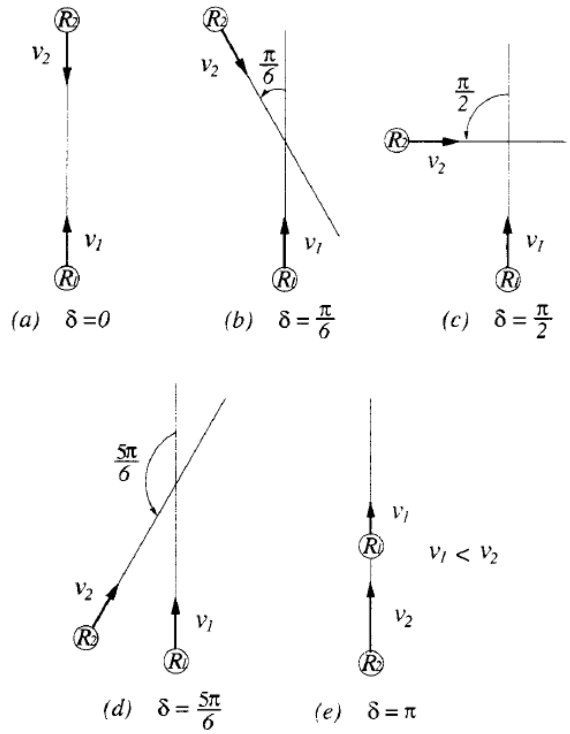

Figure 3.14. Possible collision situations for two robots. Fujimori et al. [116]

Where P_{12} is the crossing point between robots 1 and 2, and $\overline{R_i P_{12}}$ is the distance of the robot *i* to the crossing point. For the case when the robots are moving along the same line, they define the priority as $w_i = v_i$. The collision avoidance angle (direction of the robot) and velocity are given by:

$$\theta_{ci} = \theta_i + \Delta\theta_{ci} \tag{3.9}$$

$$v_{ci} = v_i + \Delta v_{ci} \tag{3.10}$$

Where $\Delta\theta_{ci}$ and v_{ci} are an additional angle and additional velocity, which could be negative, for cooperative avoidance. Authors provide formulae to compute these variables as an interpolated value between zero change and the maximal angle or velocity, respectively, for the robot with higher priority; and for the robot with lower priority this value varies from zero to the minimal value of angle or velocity, respectively. Complete interpolation formulae can be found in [116].

Finally, to extend the solution for more than two robots, authors define the robot priority as the set of priorities between this and all others:

$$w_{ij} = \frac{v_i}{\overline{R_i P_{ij}}} \tag{3.11}$$

Where $w_{ii} = 0$, if P_{ij} does not exist, $w_{ij} = v_i$, and if there is not collision detected, $w_{ij} = 0$. Therefore, the additional angle and velocity for each robot is given by:

$$\theta_{ci} = \frac{\sum_{j=1}^{n} w_{ij}\theta_{cij}}{\sum_{j=1}^{n} w_{ij}} \tag{3.12}$$

$$v_{ci} = \frac{\sum_{j=1}^{n} w_{ij}v_{cij}}{\sum_{j=1}^{n} w_{ij}} \tag{3.13}$$

Where θ_{cij} and v_{cij} are the additional angle and velocity for robot R_i with robot R_j, respectively.

To solve a collision scenario with several vehicles, Düring and Pascheka [94] define a series of steps:

1. A trajectory conflict is detected.

2. A message is broadcasted to initiate a decentralized decision-making algorithm.

3. A set of available maneuvers is taken from a pre-defined set of possible evasive maneuvers.

4. Maneuver-related information is exchanged between agents.

5. Each agent decides its course of action by choosing the best cooperative combination of maneuvers.

A maneuver is defined as an initial state and the desired changes in velocity and lateral position (Δs, Δd). Changes in speed come from a set of p discrete speed changes $\{\Delta s_k$, k = 1,2, ... $p\}$; and changes in lateral position are defined as a set of three possibilities: $\Delta d \in \{\Delta l, 0, \Delta r\}$, i.e. left-changing, lane-keeping or right-changing maneuver. Additionally, each maneuver in this pre-defined set has a cost value depending on a cost function; they explain that it could consider energy efficiency, time efficiency, driving comfort and safety. In their simulations, they used the cost of a maneuver as the weighted sum of longitudinal and lateral mean-square accelerations of the corresponding trajectory. Hence, the utility of a maneuver M is denoted as any reduction of cost with respect to the function:

$$c(M) = \frac{1}{T}\int_0^T \frac{\ddot{s}^2 + w_d \ddot{d}^2}{1 + w_d} dt \tag{3.14}$$

Where $\ddot{s}^2$ and $\ddot{d}^2$ are the longitudinal and lateral part of the maneuver, respectively, w_d is a weighting factor and T is the maneuver duration.

With a different perspective, Hafner et al. [62] present a cooperative driving solution for collision avoidance; this research uses V2V communication to coordinate, in a decentralized manner, the avoidance of collisions when two vehicles reach an intersection. The authors' approach is interesting because their avoidance algorithm is based in a control system that formally encodes a set of vehicle configurations that should be avoided in order to comply with a no-collision requirement, i.e. all speed and position configurations that could derive in a vehicle collision; they call this, the *bad set*. Later, based on current vehicle dynamics, the system computes a capture set, which corresponds to all vehicle configurations that lead to the *bad set* independently of any acceleration or braking action. Then, it calculates the acceleration/brake needed to keep the system state outside of the capture set at all times. This ensures that the vehicles will never be in a collision situation.

Experiments were performed on a real vehicle, in a collision avoidance test environment at the Toyota Technical Center, USA. According to their results, the implementation guarantees that the control system will act only if a collision might take place, in which case it will always be avoided. Their proposal works only for two vehicles approaching an intersection, and this condition keeps computing the states set relatively easy. Nevertheless, according to the authors, the computations of the sets are very demanding; so, in a more realistic situation with more than two vehicles, this computation and the generation of the corresponding distributed solution, might be excessively expensive and time consuming for a system requiring time responses in the order of hundredths of a second.

Van den Berg et al. [117] propose a velocity based approach for collision avoidance for holonomic robots: the optimal reciprocal collision avoidance (ORCA); it considers the velocity vector of surrounding robots and introduces a concept of velocity space. This space refers to planes of reachable regions for a time window τ, among those regions the system marks some of them as forbidden due to the presence of other robots in them; thus, leaving other space regions with valid velocities for the robot. Authors define a velocity obstacle VO as the set of all relative velocities of a robot A with respect to B that will result in a collision at some moment before τ. Figure 3.15(a) shows the robots A and B with radius r_A and r_B at points P_A and P_B, respectively; Figure 3.15(b) is a geometric interpretation of the velocity obstacle for robot A.

Therefore, the velocity obstacle is a truncated cone with apex at the plane origin, with legs tangent to the circumference of radius $r_A + r_B$ centered at P_B-P_A. The length of the region occupied by the cone depends on the value of τ. Finally, the system defines a set of collision avoidance velocities for robot A, when robot B selects its velocity from a set V_B, as the complement of the Minkowski sum of VO and V_B. From this valid set, the system selects the optimal velocity using linear programming. Refer to [117] for full solving equations.

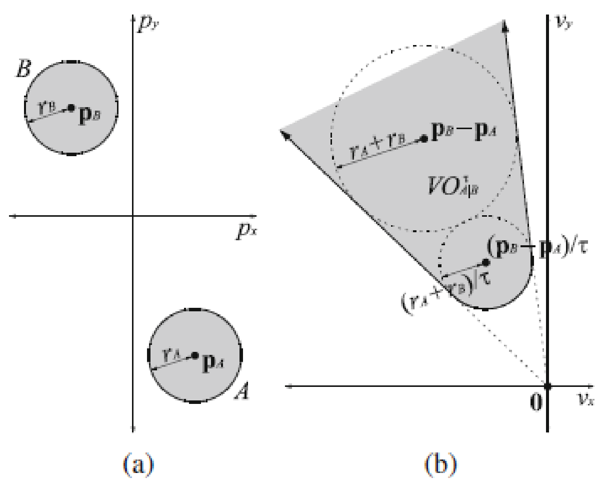

Figure 3.15 Velocity obstacle for robot A relative to robot B [117]

Alonso-Mora et al. [118] use the concept of ORCA [117] to build a collision avoidance system for multiple car-like robots. Authors extend the collaborative collision avoidance method by integrating the non-holonomic constraints and the kinematics of a vehicle. They use the simplified bicycle kinematics to model a vehicle with fixed rear wheel and a steerable front wheel. They define the vehicle coordinates as $q = (x, y, \theta, \varphi)$, where (x, y) represent the position of the rear wheel, θ is the car's orientation and φ is the steering angle. Then, they provide the kinematic formula that rules the vehicle movement as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\theta \\ sin\theta \\ tan\varphi/L \\ 0 \end{bmatrix} v1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v2 \tag{3.15}$$

Where L is the car length, $v1$ and $v2$ are the driving and steering velocity inputs, respectively. Authors also bound velocities, acceleration and steering angle to some predefined values. Following the velocity set concepts of ORCA, each robot follows these steps:

- Obtain a preferred velocity towards the goal v_i^{pref}.
- An extended radius is computed as:

$$er_i = r_i + min_j(\varepsilon_i, \frac{d(i,j) - r_i - r_j}{2}) \tag{3.16}$$

Where $d(i, j)$ is the distance between robots i and j.

- Robots within a radius of d_{max} are considered for the computation of the collision-free velocities set $ORCA_i^{\tau}$.
- The collision-free velocity is computed as the closest to the preferred velocity:

$$v_i = argmin\|v - v_i^{pref}\| \tag{3.17}$$

The collision-free velocity must also be a member of the $ORCA_i^{\tau}$ set and it must be a valid velocity for the vehicle given current speed, velocity and steering angle.

Authors explain that trajectories generated by their proposal are collision-free at each time-step; this is guaranteed by keeping the distance between two robots greater than the sum of their radii. However, in crowded environments, convergence to goal destinations for all robots is not guaranteed in a reasonable time; mainly because of the occurrence of deadlocks while computing the collision-free velocity.

Rodrigues et al. [63] present an approach aimed to enable a distributed cooperative decision-making process for vehicles crossing an intersection, while maintaining a collision-free environment. Among the assumptions of this research are: the vehicles are autonomous, their paths are known, and they do not change. They consider the intersection crossing as a special case of a scheduling problem; thus, the objective is to optimize the access order to the shared resource (the intersection). Thus, this implementation transforms the problem to a control scheduling one; in which vehicles solve their own crossing schedule problem sequentially. Similar to [62], authors in [63] propose the computation of a *critical set* which holds, for a single vehicle, all displacements that could lead to a collision. Thus, a collision is considered to occur if at least the states of two cars are in the respective sets at a same time instant.

Using reachability analysis tools, the authors define, for each vehicle, an attraction set as all possible state configurations that will lead it, in *n* steps, to its *critical set*. They define t_i^c as the set of all time instants where the *i*-th car's state lies in the critical set. Then, they define the time to react as the remaining time from now until the moment the car enters the critical set, and it can be computed with:

$$\Delta_i^{TR} = \left(k_j^c - n\right) - t_0, n \geq t, \forall_i \in N \qquad (3.18)$$

Where t_0 is the time when the process starts, N is the set of cars, and k_j^c is:

$$k_j^c = min_{j \in N}\{t_j^c\} \qquad (3.19)$$

They provide a decentralized strategy in the form of control scheduling, where vehicles, in sequence, find a local solution for an optimization problem; this will produce an expected time of arrival of the vehicle to the intersection, it will be used as a constraint for next vehicles in the sequence. The order in which vehicles compute their control strategy is given inversely proportional to the value of Δ_i^{TR}; so, vehicles that have less time to react go first. Such approach, in the words of the authors, introduces fairness to the protocol because vehicles with less time to react also have less time for large maneuvers to avoid collisions; therefore, those with higher maneuverability time go last.

Though the authors say that the proposal is scalable, this was not demonstrated as the simulations involved only three vehicles; also, being this an optimization problem, an increase in the number of vehicles will demand more computing time to find a solution, hence, less time to react to any given situation.

With an increasing number of vehicles with autonomous or semi-autonomous capacities, such as collision avoidance, lane change or take over, and manufactured by different companies, it is of major importance that all of them agree on what is the situation around them when they share roads. Even if they might have different approaches to achieve their goals, they must have a common understanding of the environment; misinterpretations of security messages could lead to very dangerous situations. Eigner and Lutz [119] propose a collision avoidance application that uses context information based on an ontology and an inference engine. Vehicles are considered as rectangles and they are simulated as members of a VANET to exchange location, speed and acceleration data; this information is stored in the ontology using OWL. To compute if a collision might occur, they built a system of non-linear inequalities that describes the possible overlapping of the rectangles that contain the vehicles. A Fourier-Motzkin elimination process is used to solve the system; if a value $t \geq 0$ exists, then a collision will occur. The main contribution of this work is the ontological approach applied to vehicle networks, street information and specially to collision avoidance; this approach could be seen as a foundation for interoperable systems that will replace current fleet of vehicles in the streets. However, in order to use it in more realistic tests, additional physical variables must be included, as well as a more representative physical vehicle model; also, the process of input and output of knowledge in the individual vehicles is not presented, which is a requirement for the definition of a cognitive agent.

3.5 Summary

In this chapter we made an extensive review of the state of the art, considering the subjects that are close to our research's interest themes. Modelling a connected or autonomous vehicle was presented according to the base concepts and the de facto standards in the literature. In the same way, we made a detailed discussion of the works that study the cognitive component of these vehicles; intelligent agents have shown to be one of the knowledge areas with more strength in this sense. The analogy of modules between an autonomous vehicle and an intelligent agent is evident: sensors, environment processing, and actuators; according to what was described, the authors differ on the way to organize the cognitive behavior required for decision making.

The use of intelligent agents with communication capabilities allows the creation of multi-agent environments, where several of them can collaborate to execute a common task. The way a group of autonomous vehicles can perform as a multi-agent environment is still one of the challenges to pursuit, especially for road safety situations. Even if there are studies focused on collision avoidance, it does not exist yet a consensus on how various vehicles on the road can collaborate to achieve this goal. In the next chapter, we will make a comprehensive analysis of the state of the art, its main contributions and deficiencies, which will serve as a guide to orient this research.

Chapter IV
Objectives

4.1 Introduction

In Chapter 3 we presented the state of the art of the research topics related to our work; we have also mentioned that the research axles leading this study are: vehicle modeling, cognitive modeling, collision avoidance and cooperative driving. The review has given us a global impression of current focus of the literature. In this chapter, we will summarize and discuss major contributions and limitations that will help us identify the direction of our research in the form of objectives.

4.2 Analysis of the state of the art

In order to organize the analysis of the explored state of the art, in this section we start by considering the criteria that is most relevant to the present work: the physics model, the cognitive and collective behavior, the use of a centralized or a distributed approach, the use or not of a knowledge representation, and the integration of configuration sets for valid and invalid collision avoidance alternatives. With this basis, in Table 4.1 we make a recap of the related works; this will support a founded analysis for the detection of weaknesses and lacunas that need to be explored.

Table 4.1 Literature organized by relevant criteria

Work	Physics model	Cognitive behavior	Collective behavior	Centralized/ Distributed	Knowledge representation	Config sets
Fiosins et al. [88]	Y	Y	N	D	N	N
Jimenez et al. [102]	Y	N	N	D	N	N
Guo et al. [108]	Y	N	N	D	N	N

Wu et al. [56]	N	N	Y	C	N	N
Lee and Park [57]	Y	N	Y	C	N	N
Maile et al. [61] [112]	Y	Y	N	C	N	N
Hafner et al. [62]	Y	N	N	C	N	Y
Burmeister et al. [73]	N	Y	N	D	N	N
Albus [87]	N	Y	N	D	N	N
Wahle et al. [89]	N	Y	N	D	N	N
Eigner and Lutz [119]	N	Y	N	D	Y	N
Caveney and Dunbar [113]	N	Y	N	C	N	N
Wang et al. [95]	N	Y	N	C	N	N
Abbas-Turki et al. [110]	N	N	Y	C	N	N
Ahmane et al. [111]	N	N	Y	C	N	N
Rodrigues et al. [63]	N	N	Y	C	N	Y
Huang and Tan [115]	N	N	N	C	N	N
Qu et al. [100]	N	N	N	C-D	N	N
Regele [74]	N	N	N	D	Y	N
Patron et al. [75]	N	N	N	D	Y	N
Ejigu et al. [76]	N	N	N	D	Y	N
Chen et al. [77]	N	N	N	D	Y	N
Wannous et al. [78]	N	N	N	D	Y	N
Chen et al. [79]	N	N	N	D	Y	N
Alonso-Mora et al. [118]	N	N	N	D	N	Y
Van den Berg et al. [117]	N	N	N	D	N	Y

From Table 4.1, we can detect a lack of studies, in the field of autonomous vehicles, that benefit from the use of information sharing capacities to create cooperative approaches in order to avoid collisions. Solutions that generate isolated maneuvers to avoid accidents might not be able to prevent them in all possible scenarios; therefore, a cooperative behavior is required to reach a broader range of situations. The synergy of the organized and combined actions of multiple vehicles can outperform the separated efforts of the many. Presented literature that includes collective behavior is mainly based on a centralized entity that decides and controls the avoidance maneuvers. We have detected a significant absence of distributed collision avoidance approaches that use not only location data but also the intention of nearby vehicles in order to generate reaction plans; collision avoidance systems without such intention information rely only on on-board sensor data and therefore produce solutions with incomplete evidence.

Essentially, current state of the art is limited to the use and test of scenarios with one or two equipped vehicles ([61], [62], [88], [89], [102] [106], [108], [99] and [115]). Although decision making in almost all of these solutions is distributed, authors consider them also as cooperative just for the fact that they share location and intention information, but not because they find a global plan in which all agents have a role.

Approaches that produce cooperative solutions to avoid a collision generally use a centralized algorithm to find this collective solution ([57], [63], [100], [110] and [111]); as was explained in section 3.3 when discussing the modeling of cooperative capacities, centralized solutions are easier to implement, but they suffer of the bottleneck problem when too many messages are in place.

Although the works of [56], [57], [109], [110] and [113] consider several cars for their tests, they must be all connected and to find a solution they use centralized processes. One major constraint in the presented state of the art is the use of scenarios only with connected cars; such configuration does not reflect the reality of the adoption of connected and autonomous vehicles. In an early stage, the integration of vehicles with this technology and with cooperative capabilities will be progressive; thus, there is a research opportunity in the field to experiment with both automated and human-driven vehicles sharing a common environment.

None of the reviewed works provides a visual approach to evaluate the effectiveness of a collision avoidance solution; thus, researchers are limited to evaluate and analyze output data and are not able to actually watch how a resultant maneuver is executed and how it avoids a collision. Without the possibility of visual exploration, other works provide distributed solutions for multiple vehicle scenarios, it is the case of [87], [100] and [115]. Also, [63], [116] and [117] are distributed approaches for collision avoidance with an additional component of restraining sets.

Most of the works use algorithmic methods to detect a dangerous situation and to react to it; nonetheless, some authors state that solutions should be considered as cognitive if they include knowledge and context elements. In this sense, even though there are researches using ontologies ([74], [75], [76], [77], [78] and [119]) for the ITS, they only consider static data of the vehicles and their environment; also, they are restricted to store knowledge and to deduce from it if a collision might take place, none of them uses that knowledge to produce actions to avoid detected dangerous situations. They are also limited to consider two or three vehicles in the scenarios, as well as to produce numerical results of static scenarios, without graphical simulations and experimentation to observe results or computing time.

While exploring cognitive approaches, it is worth to mention that an ontology-based kb allows the researcher to define explicit relationships among the data. We assume in this research that an intelligent agent drives the vehicle, and therefore the cognitive capacities are related to the agent while perceiving the vehicle's environment and acting over it; therefore, we will often refer to agent and vehicle as the same entity. So, the vehicle, modeled as an agent, could take advantage of a knowledge base, in the form of a hierarchy of entities and relationships, to have insights of what is going on in its environment and to predict the occurrence of collisions.

Another pertinent contribution of the reviewed state of the art is the integration of the time dimension into the cognitive analysis ([78] and [79]), along with the space dimension; this has given the opportunity to determine, via logic cognitive rules, if two entities share the same spot at the same moment. The incorporation of the time dimension helps to produce what authors have named the situation awareness; intelligent agents can know what is the current configuration of its surroundings, and what it will be in the near future, thanks to available information related to the presence of entities such as vehicles or pedestrians, their location, orientation, speed and intentions.

An autonomous vehicle that is not aware of its environment cannot determine which are the necessary maneuvers to execute in face of risky circumstances; moreover, if the vehicle agent is not "conscious" of the time, and does not consider the duration of each action in a maneuver, then it will not be able to properly generate avoidance plans, constrained to a global completion time and to the plans of other actors in the surroundings.

Revised literature related to agent communication agree that, even if agents can be of different type, they must share a common communication protocol, as well as a common understanding of the information regarding their environment. Most of the works use an architecture of two or three layers ([73], [75], [88], [89]), which is consistent with the time constraints related to collision avoidance capabilities; having fewer phases with well-defined and distinguishable responsibilities helps to execute a faster decision-making process. Time constraint is a topic of major relevance, as was exposed in the previous chapter, in a regular collision scenario several vehicles are moving; consequently, the detection of the danger, communication with others, consideration of available options, decision making and maneuver execution must be all completed while vehicles are still moving and before the collision occurs.

The inclusion of the configuration sets proposed by [63], [116] and [117], is an interesting contribution of some cooperative avoidance proposals, the idea of separating undesired situations from desired ones provides a way to not just avoid a collision but also to avoid a dangerous situation from ever appearing. Works in this area show positive results with zero collisions even in environments with a high density of agents. Another important contribution is the classification of collision situations made in [116]; this helps to define a single avoidance strategy for each type of collision situation, simplifying the decision-making process of the agent.

The conception of a new cognitive model to store knowledge of urban environments, vehicle control data, vehicle intentions, accidents and traffic information, as well as to provide a unique common structure where every actor interprets the information in the same way, is a requirement as a basis for the interoperability of future developments in the field.

4.3 Objectives

To collaborate with the research on the limitations detected in the literature, and to propose a novel solution to the collision avoidance problem, integrating a cooperative approach with cognitive agent behavior, we have defined the following research objectives:

- Incorporate context awareness in vehicles in the form of a knowledge base representing their current environment and intentions.
- Design a cognitive structure for a vehicle agent to detect if a collision situation might occur by using environment information and intention of neighbouring cars.
- Generate cooperative maneuvers for collision avoidance based on context in scenarios involving more than two vehicles.

Chapter V
Methodology

5.1 Introduction

As was mentioned in chapter 2, even if semi-autonomous features are currently available for commercial vehicles, full autonomy is still in a testing phase; by considering the scenarios described in chapter 2, it is clear that there still exist challenges to overcome before the technology is adopted for a majority of the vehicles on the roads. Situations where the cameras of an autonomous vehicle are not able to detect its environment, due to visibility obstructions or because of distance constraints could become a serious danger to the occupants and to pedestrians.

The discussion made in chapter 4 allowed us to analyze the direction of the approaches related to our study subject, as well as their limitations; we have detected that research efforts are mainly focused to study isolated autonomy capabilities, i.e. vehicles that do not collaborate. Moreover, in most of the reviewed literature, the configuration of scenarios is homogeneous, in the sense that it considers only connected or autonomous vehicles in the tests and simulations; the use of numeric results alone, without visualization capabilities, is another common limitation of the literature that reduces the ability to analyse collision avoidance strategies.

In this chapter, we will explore the methods and technologies that can help us achieve the objectives defined in the previous chapter. By taking advantage of communication capabilities in vehicles, they will be able to share their intentions, as well as relevant environment information, in order to collaborate with neighbouring cars; such collaboration is a distributed decision-making process that generates collective avoidance maneuvers considering current context and the intention of others.

The present research intends to resolve exposed lacunas in the field of vehicle collision avoidance by proposing a cognitive approach; the objectives proposed in chapter 4 serve as a guide to accomplish such proposal. In order to have several vehicles, of different types, performing avoidance maneuvers as projected in our first objective, we will begin by defining the foundations of the vehicle's cognitive model; which will be needed to store context knowledge.

The remaining of the chapter is organized as follows: first we will expose the assumptions of our proposal; then we present the design of a knowledge base to provide context awareness to vehicles, by representing the status of the environment as well as the vehicle's intention; later, we will explain how the elements of the context can be used to detect the possible occurrence of a collision; consequently, the elements of the cognitive collision avoidance system are presented. We finalize by describing the details of the implementation of a 3D simulation tool that will serve to validate the proposed approach.

5.2 Assumptions on the environment and the vehicle

The environment considered in our approach is that of a highway with at least three lanes, and the host vehicle (v*0*) with collision avoidance capabilities going on the center lane as depicted in Figure 5.1. The vehicle *v0* has proximity sensors that allow it to compute the distance to close obstacles, such as other vehicles. Also, a localization system and a map of the road. The vehicle also has communication capabilities with other vehicles and with the connected infrastructure. The operation of these components is out of the scope of this research, therefore in our vehicle model we consider that data provided by them is true and valid.

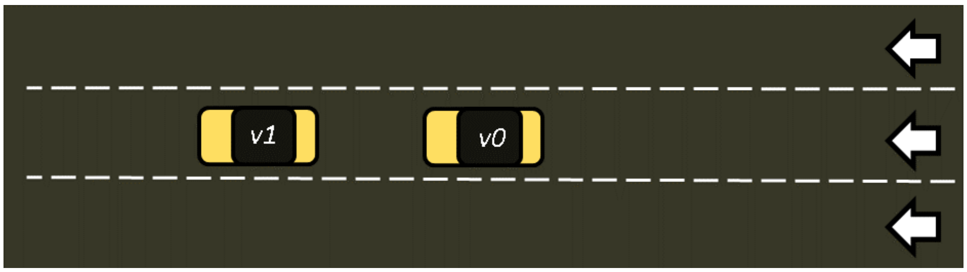

Figure 5.1 Highway environment considered for the research

Similarly, a decision-making system is incorporated in *v0* giving it the capability to recognize the risk of collision with *v1* and to trigger an avoidance maneuver, if needed. The focus of this research is the design and development of such decision-making system; in this chapter, we explain its components and their integration.

5.3 Incorporate context awareness in vehicles in the form of a knowledge base representing their current environment and intentions

As discussed in chapter 2, on-board sensors provide the vehicle with information regarding its close environment; however, thanks to communication capabilities a vehicle can also be aware of what are the intentions of its neighbors, if they are going to turn, to brake or if they are in an emergency. The combination of the information coming from these two sources is what we define here as the context; knowing what other vehicles are planning to do next is an important input for our collision avoidance proposal since it can be used to produce a cooperative strategy.

We planned as our first objective to provide context awareness to vehicles by designing a knowledge base that will contain information about the vehicle's environment and its current intention. Thus, when the vehicle faces a dangerous situation, the decision-making process will be based on the information stored in this knowledge base; then, instead of making isolated decisions, the vehicle can resolve to execute a safe maneuver based on what its neighbors are planning to do. As was exposed in chapter 4, literature is limited on the description of how a vehicle agent stores and interprets the environment information; which is essential, because the mutual understanding of agents depends on that.

Therefore, we propose to design an ontology as the agent's knowledge base; it defines the entities of a street environment and the relations between them. Figure 5.2 depicts our proposal of an ontology for a cooperative collision avoidance capable vehicle; it is designed to store the data of the streets present in the scenario, the current and the future location of the vehicle, and the location of neighbouring vehicles.

In this ontology, we relate basic entities of a vehicular environment, these are: the streets, the lanes and the cars. Additional entities serve to store information of current location of the vehicle, as well as to identify where the vehicle will be in the near future. Furthermore, time interval entities, based on the work of Wannous et al. [78], complement the spatiotemporal elements required to determine if two or more vehicles can be at the same location at the same time.

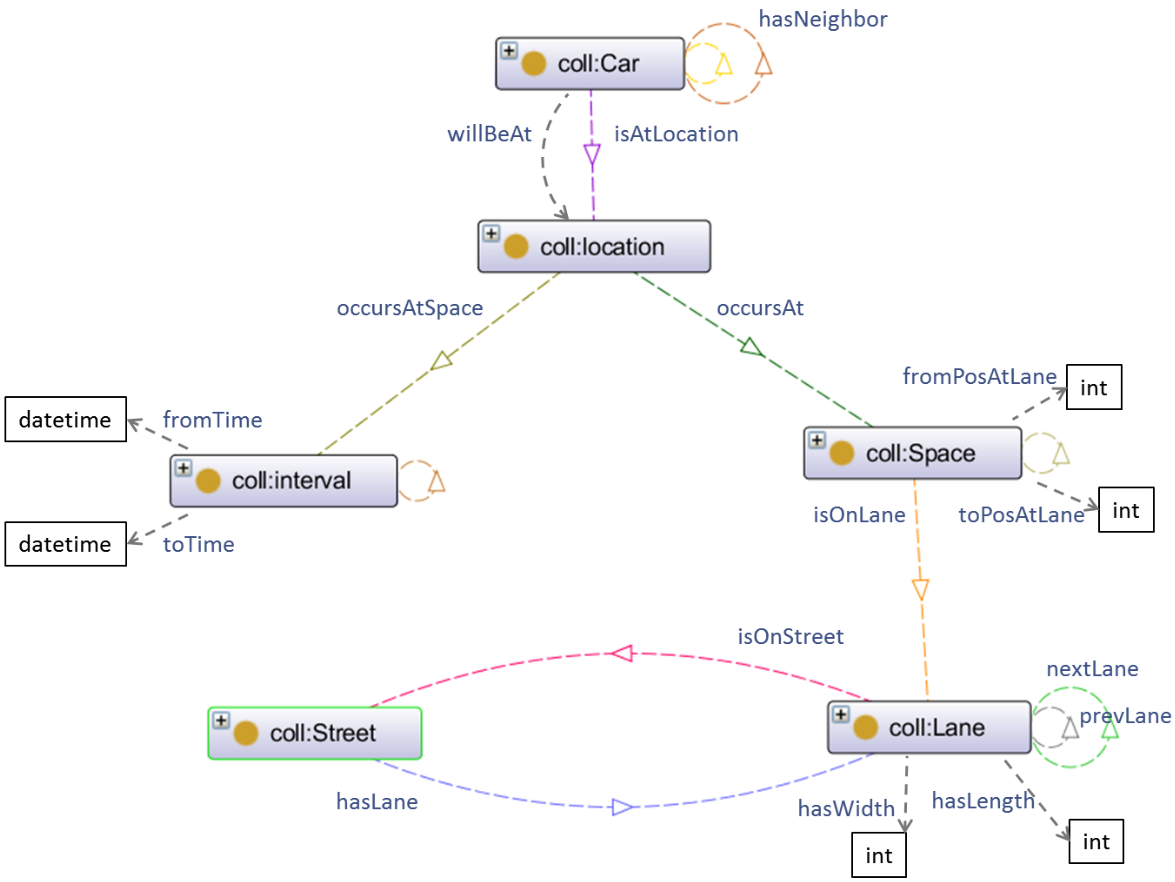

Figure 5.2 Ontology proposed as knowledge base for the agent's cognition

A description of the entities presented in Figure 5.2 follows:

Street: instances of this entity describe the streets currently stored in the knowledge base.

Lane: this entity refers to the different lanes in a single street. Main attributes of this entity are the length, width (*hasWidth* and *hasLength*).

Space: this entity defines a portion of a lane that can be occupied by a car. Main attributes of this entity are the start and finish position of the space on the lane (*fromPosAtLane* and *toPosAtLane*)

Interval: this entity defines a time interval. It has two *datetime* attributes: the start and end of the time interval (*fromTime* and *toTime*). Instances of this entity determine when the vehicle is at a certain street space.

Location: this entity is composed of two object attributes, a space and an interval. Thus, an instance of location indicates where and when the vehicle is.

Car: it is the entity to describe the vehicles in the knowledge base.

The strength and usefulness of an ontology is based on the relations of its entities and, in order to deduce knowledge from it, it is necessary to have logic relations between the entities. So, according to Figure 5.2, there are several relations that will allow us to detect the risk of a collision; the description of those relations follows:

The *isOnStreet* relation indicates that a *Lane* instance is on a *Street*.

The *hasLane* relation indicates that a *Street* instance has *Lanes* that compose it.

The *isOnLane* relation indicates to which *Lane* a *Space* instance belongs.

The *prevLane* and *nextLane* relations indicate which lanes are behind and ahead of a *Lane* instance. They relate a *Lane* instance with other two.

The *isAt* relation indicates at which *Location* a particular vehicle is.

The *hasNeighbor* relation in the *Car* entity states that a car instance can have another car instance as a neighbor. In fact, this relation supports a car having more than one car in the vicinity; the consideration of a car as a neighbor would depend on the range of the communication antenna.

Using the same logic of *isAt*, the *willBeAt* relation indicates at which *Location* a vehicle will be in the future if current vehicle state remains in time. The distinction between where the vehicle is right now, and where it will be, only depends on the time interval associated to the *Location* instance.

5.4 Design a cognitive structure for a vehicle agent to detect if a collision situation might occur by using environment information

The ontology presented in the previous section serves as a starting point to define a cognitive approach to detect the possibility of a collision; which is the second objective of the present research. Given that instances of the *Location* entity are spatiotemporal objects, we can use them to detect if a collision might take place; querying the ontology for two or more *Car* instances that have the same *Location* instance as their *willBeAt* property, will return either *void* or a set of colliding cars.

To achieve this objective, we have defined a series of logic predicates. First, the *overlap* predicate to detect if two given *Interval* instances overlap in time; the *spaceOverlap* predicate to find out if two given *Space* instances refer to the same spot in a street; and finally, the *collisionWith* predicate to recognize if two different *Car* instances will be occupying the same street space at the same time. These predicates are presented in Table 5.1.

Table 5.1 Logic predicates to detect a collision between two *Car* instances

Overlap predicate
interval(?I1) ^ **interval**(?I2) ^ DifferentFrom (?I1, ?I2) ^ **fromTime**(?I1, ?from1) ^ **fromTime**(?I2, ?from2) ^ **toTime**(?I1, ?to1) ^ **toTime**(?I2, ?to2) ^ greaterThanOrEqual(?to1, ?from2) ^ greaterThanOrEqual(?from2, ?from1) **-> overlap**(?I1, ?I2)
spaceOverlap predicate
Space(?sp1) ^ **Space**(?sp2) ^ **isOnLane**(?sp1, ?lane1) ^ **isOnLane**(?sp2, ?lane2) ^

```
hasLaneId(?lane1, ?lid1) ^ hasLaneId(?lane2, ?lid2) ^
notEqual(?lid1, ?lid2) ^
fromPosAtLane(?sp1, ?fromPos1) ^ toPosAtLane(?sp1, ?toPos1) ^
fromPosAtLane(?sp2, ?fromPos2) ^ toPosAtLane(?sp2, ?toPos2) ^
greaterThanOrEqual(?toPos1, ?fromPos2) ^
greaterThanOrEqual(?fromPos2, ?fromPos1)
-> spaceOverlap(?sp1, ?sp2)
```

collisionWith predicate

```
Car(?c) ^ Car(?c2) ^ DifferentFrom (?c, ?c2) ^
willBeAtLocation(?c, ?l1) ^ willBeAtLocation(?c2, ?l2) ^
occursAtSpace(?l1, ?sp1) ^ occursAtSpace(?I2, ?sp2) ^
occursAt(?l1, ?int1) ^ occursAt(?l2, ?int2) ^
overlap(?int1, ?int2) ^ spaceOverlap(?sp1, ?sp2)
-> collisionWith(?c, ?c2)
```

An autonomous vehicle that stores cognitive data in the knowledge base would be able to use the proposed functions to anticipate the future occurrence of a collision situation. When such situation is detected, the autonomous vehicle must perform an avoidance maneuver. In the following section, we explain our proposal of a system that generates avoidance maneuvers using context information.

5.5 Generate context-based cooperative maneuvers for collision avoidance considering vehicles' intentions in scenarios involving more than two vehicles

Our third objective, as defined at the end of chapter 4, is to generate context-based cooperative collision avoidance maneuvers; for that, we are proposing a new approach characterized by context awareness in which vehicles share knowledge about their close environment as well as their intentions. In this section we explain that, depending on the particular collision situation the vehicle is facing, our proposal only considers information that is relevant for the generation of an avoidance maneuver.

5.5.1 Elements for collision avoidance

To avoid a collision, in any case, a car has the option to change its longitudinal speed, via the accelerator or the brakes, to change its orientation, or a combination of both. The gas throttle and the brakes change a vehicle's acceleration in opposite directions, and a valid situation on the road will not require pressing both pedals at the same time. Thus, to simplify the elements of our proposal, we use a single variable to indicate how much the vehicle is accelerating or braking; we call it *gasAmount*:

$$-1 \leq gasAmount \leq 1 \tag{5.1}$$

If *gasAmount* is greater than zero, it refers to how much the accelerator is pressed, if it is lower than zero then it refers to how much the brake pedal is pressed; the boundaries of the variable refer to fully pressing the accelerator, in the positive case, or the brakes, in the negative case.

Similarly, to set a new heading for the vehicle, it must turn the steering wheel, left or right. We call this *steerAmount* and its value is set in radians:

$$-3\Pi \leq steerAmount \leq 3\Pi \tag{5.2}$$

Where a negative value of *steerAmount* indicates that the steering wheel must be turned left, and a positive value represents a right turn of it. The maximal rotation in either way is 3Π, i.e. one and a half turns of the steering wheel. Then, we can define that an action of the vehicle is composed by a *gasAmount* and a *steerAmount*:

$$Ac = (gasAmount, steerAmount) \tag{5.3}$$

5.5.2 Collision risk

In chapter 3 we explained that the Time To Collision (TTC) is a quantitative value used to determine the risk of a collision, and the various ways to estimate it. In this research, we are using the TTC explained by Milanés [105] which follows equation (5.4).

$$TTC = \frac{D_{cur} - D_m}{S_0 - S_1} \tag{5.4}$$

Where S_0 and S_1 are the speeds, in meters per second, of the host vehicle ("*v0*" in our case), and the preceding vehicle ("*v1*"), respectively. D_{cur} is current distance between cars, and D_m is a minimum acceptable distance when both vehicles stop. Considering the recommendations of Lee et al. [104] and Milanés [105], for the purposes of this research, we state that a collision risk exists when the TTC is 2 seconds or less.

5.5.3 Context awareness and decision making

In this section, we present the decision-making process that will be triggered by an autonomous vehicle when a collision risk appears.

We propose here that vehicles perform the collision avoidance in two stages. The first one is used for situation awareness, and the second stage generates a maneuver to avoid the collision. The situation awareness allows us to minimize the number of parameters needed to generate the evasive maneuver, thus simplifying the process. So, given the current context of the vehicle, the first stage will determine in which way the evasive maneuver should be performed. The information that is relevant for context awareness and collision avoidance is that of the vehicles going on the contiguous lanes, we depict it as “Right sector” and “Left sector” in Figure 5.3.

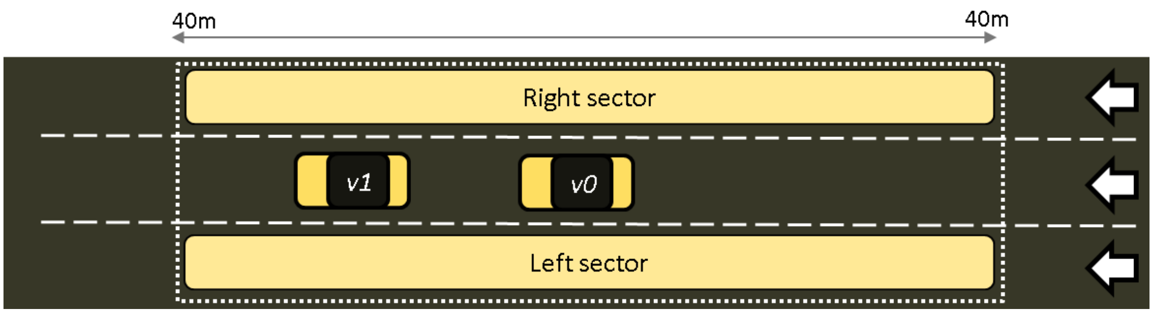

Figure 5.3 Zones of interest for context awareness in collision avoidance

The context information of interest in Figure 5.3 is related to the existence, or not, of a circulation lane on the left and the right, the speed, distance and intention of the closest vehicle in the mentioned relevant sectors, as well as of the vehicle in front (*v1*). The context of *v0* reaches away up to 40 meters as depicted by the dotted rectangle in Figure 5.3, vehicles beyond this distance are not considered as part of the context of the vehicle. The information is stored in the vehicle’s knowledge base and used later as input of the first stage of our proposed system.

As part of our proposal, we take into account the intention of the vehicle when it is avoiding a collision. Therefore, when a vehicle is changing lanes, it informs its neighbors about its intention. Then, in a form of cooperation, we state here that vehicles receiving this information will consider that the remote vehicle is already in the target lane. Thus, if another vehicle is also on the target lane, it will trigger the system and apply its own avoidance maneuver.

Using context information, the vehicle can decide which avoidance direction minimizes the collision risk. It can either change lane to the left, to the right, or brake remaining on the same lane. As mentioned in chapter 3, authors use metaheuristics, neural networks or fuzzy systems for optimization and decision-making. Metaheuristics are mostly considered when dealing with big solution spaces and require long computation times, which is not desired in a collision situation. Although neural networks, and other supervised machine learning techniques, have fast response times, they require lots of data to train the models. Lastly, fuzzy systems don't have the mentioned disadvantages and are traditionally used for decision-making purposes. So, to accomplish our goal of context-awareness, at the first stage of our collision avoidance proposal we use a fuzzy system. Lee [120] mentions that fuzzy systems are very useful when the sources of information are interpreted qualitatively or with uncertainty. Such systems are supported by a group of fuzzy sets, which represent qualitative values for the inputs, also known as *linguistic variables*. A fuzzy set is characterized by a membership function, which takes values in the interval [0, 1]. This function indicates the degree of membership of a linguistic variable to a particular fuzzy set; it should be noticed that fuzzy membership functions of different fuzzy sets can overlap each other.

A *fuzzification* process in a fuzzy system uses the membership functions to map scalar inputs into linguistic values for the linguistic variables; the linguistic values can be viewed as labels of the fuzzy sets. A decision-making logic is built as the core of the fuzzy system in the form of *linguistic description rules*; these rules consider linguistic values of the inputs to produce linguistic values for a set of output variables. They have the form:

```
IF (set of conditions are satisfied) THEN (set of consequences are true)
```

The set of conditions (or antecedents) and the consequences (or consequents) are associated with a group of linguistic variables. The linguistic variables associated to the consequent of a rule are also referred as its output. This output comes in the form of a linguistic value that must be *defuzzified*.

So, we associate the context of the vehicle to a set of linguistic variables as input of the fuzzy system in the first stage; there are 6 linguistic variables that characterize the context. In Table 5.2, we describe the variables we propose and use in this research and their corresponding fuzzy sets.

Table 5.2 Input variables of the first stage of the avoidance system

Variable	**Description**	**Fuzzy sets**
distLeft	The distance of *v0* to the nearest vehicle on the left lane.	Far, medium, close and too close.
deltaLeft	The difference in the speed of *v0* relative to the closest vehicle on the left lane.	Fast, regular, slow, away.
distFront	The distance of *v0* to the vehicle in front.	Far, medium, close and too close.
deltaFront	The difference in the speed of *v0* relative to the vehicle in front.	Fast, regular, slow, away.
distRight	The distance of *v0* to the closest vehicle on the right lane.	Far, medium, close and too close.
deltaRight	The difference in the speed of *v0* relative to the closest vehicle on the right lane.	Fast, regular, slow, away.

We believe that only the speed of *v0* is not enough to make a decision on the avoidance direction; a vehicle driving slow can still be in danger if another one is driving fast. Therefore, the relative speed between a pair of vehicles is more appropriate to this matter. We consider the relative speed as the difference of speed between a pair of vehicles; and it is computed for *v0* relative to the closest vehicle on each of the relevant zones presented in Figure 5.3. These are: *deltaLeft*, *deltaFront* and *deltaRight*, as described in Table 5.2.

The *distLeft*, *distFront* and *distRight* variables share the same membership function, which is depicted in Figure 5.4. The *deltaLeft*, *deltaFront* and *deltaRight* variables also share the same membership function, and this one is depicted in Figure 5.5.

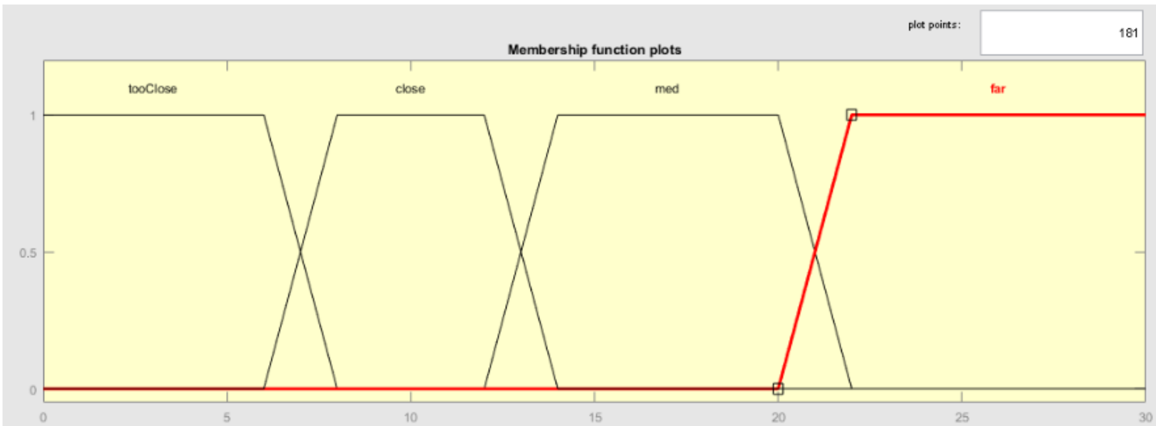

Figure 5.4 Membership function for the distance variables

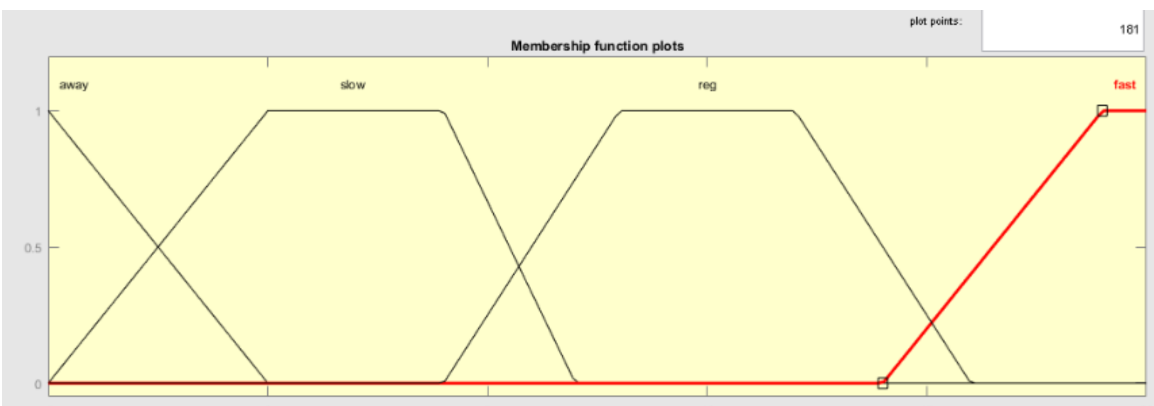

Figure 5.5 Membership function for the delta variables

The distance variables are measured in meters, and the delta variables are measured in kph. The boundaries of the fuzzy sets of the distance variables are as follows: the "too close" set has a membership function defined in the interval [0, 8], where the membership degree is maximum below 6 m, and the degree between 6 and 8 m decreases linearly down to the minimum (zero). The "close" set has a membership function defined in the [6, 14] interval, where the minimum degree of membership is at 6 m and starts to raise up to the maximum when the distance is 8 m, between 8 and 12 m the membership degree is at its peak (1), and decreases linearly from 12 to 14 m down to the minimum degree of membership. The "medium" fuzzy set has a function defined in the [12,20] interval, where the membership degree is minimum at 12 m, it raises linearly to its maximum at 14 m, it is at its peak from 14 to 20 m, and decreases linearly from 20 to 22 m. The "far" set has a function in the interval [20, ∞], where the minimum membership degree is at 20 m, it raises up to its maximum at 22 m, and it is at its peak for the rest of the interval.

Regarding the boundaries of the fuzzy sets of the delta variables, the "away" set is intended to indicate that vehicles are moving away, therefore it has a membership function defined in the interval [-5, 0], where the membership degree is maximum below -5 kph, and the degree between -5 and 0 kph decreases linearly down to the minimum (zero). The "slow" set has a membership function defined in the [-5, 7] interval, where the minimum degree of membership is at -5 kph and raises up to the maximum when the delta of the distance is 0 kph, between 0 and 4 kph the membership degree is at its peak (1), and decreases linearly from 4 to 7 kph down to the minimum degree of membership. The "regular" fuzzy set has a function defined in the [4, 16] interval, where the membership degree is minimum at 4 kph, it raises linearly to its maximum at 8 kph, it is at its peak from 8 to 12 kph, and decreases linearly from 12 to 16 kph. The "fast" set has a function in the interval [14, ∞], where the minimum membership degree is at 14 kph, it raises up to its maximum at 19 kph, and it is at its peak for the rest of the interval.

Computing the values of the distance and delta variables depends on the existence of a vehicle on the lane of *v0* or on the adjacent lane. Thus, to avoid invalid representations of the context, we consider the absence of cars on a lane by using the fuzzy sets “far” and “away” for the corresponding distance and delta variables, respectively.

To illustrate the values of input variables depending on the context, we depict in in Figure 5.6 the context of a vehicle with cars present in both adjacent lanes.

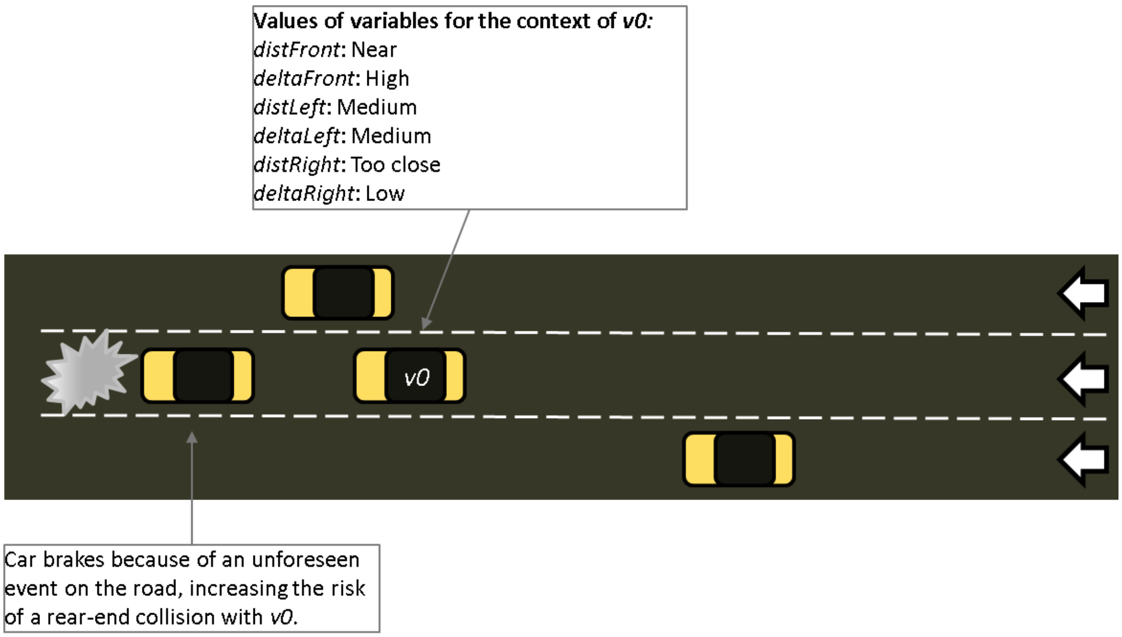

Figure 5.6 Sample situation with lanes on both sides of *v0* to illustrate values of input variables

It is clear that the way the input variables have been defined ensures that every possible combination of values represents a valid state of the context; therefore, it is not possible to have a contradictory configuration or an unfeasible situation.

As explained before, the first stage of the proposed collision avoidance process indicates the direction of the avoidance maneuver; thus, the rules of this system must have a single output (the direction). This output variable has membership functions defined with Sugeno’s singletons, as they are described in [121], [122], one for each possible avoidance direction. The possible values of the *direction* linguistic variable are described in Table 5.3.

Table 5.3 Output variable.

Variable	Description	Fuzzy sets
direction	Direction of the avoidance maneuver.	Left, center, right.

Consequently, in the second stage, we have created three fuzzy systems, one for each evasive direction possibility; only one of these systems will be called during an avoidance maneuver. Each fuzzy system receives as input only the relevant information related to it, which is used to produce the corresponding avoidance action; such action has two components: the gas amount and the steering angle (View Figure 5.7).

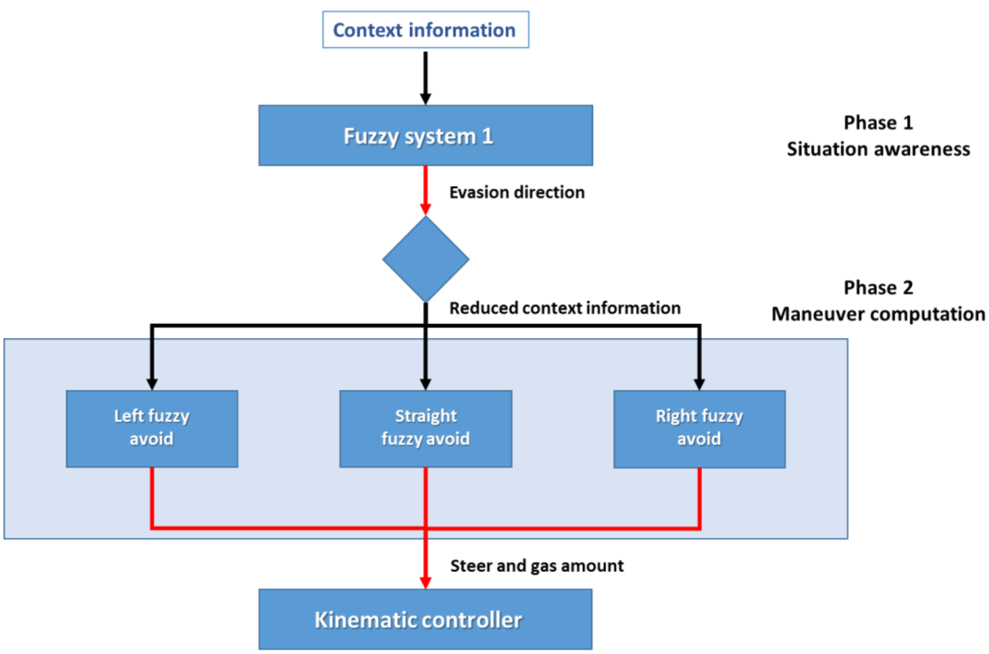

Figure 5.7 Architecture of the two-phases fuzzy collision avoidance system

Then, the vehicle can be guided using the output of the corresponding fuzzy system on the second stage, and avoid the collision. In the next section, we present the details of the systems on the second stage.

5.5.4 Maneuver control

When the vehicle knows the direction to take in order to avoid the collision, then a control must be performed over the vehicle's actuators. As explained in the previous section, we propose a fuzzy controller for straight collision avoidance and other two for side avoidance.

If the avoidance maneuver is to be performed on the same lane (straight ahead) then the "straight fuzzy avoid" controller on the second phase indicates how hard the brakes should be pressed; as the car will not change lane in this case, the steer is kept constant.

We consider the controllers of Naranjo et al. [107] as a basis for our left and right avoidance controllers. They propose two inputs: the lateral error (*latErr*) and the angular error (*angErr*); which allowed them to steer a vehicle while changing lanes. However, their work is not conceived for collision situations, and therefore there aren't any concerns about how fast the maneuver should be performed. Here, we are proposing to incorporate the linguistic variables presented in Table 5.2 as additional inputs. Therefore, the left avoidance controller has four input variables: *latErr*, *AngErr*, *distLeft* and *deltaLeft*. And the right avoidance controller has four input variables: *latErr*, *AngErr, distRight* and *deltaRight*.

According to the architecture presented in Figure 5.7, the avoidance system produces the controls for the vehicle actuators; therefore, we use two linguistic variables as output of the fuzzy systems in the second phase: *gasAmount* and *steerAmount*.

In their work, Naranjo et al. [107] concluded that, to maintain occupants' comfort during a steering maneuver, the steer change per timestep should not exceed 2.5% of the limit (which is 3Π); so, they use two fuzzy sets in their controller: one for zero steer, and the other one for steering 2.5%. However, in certain situations, an avoidance maneuver should sacrifice comfort for safety. Thus, based on their work, we include two additional fuzzy sets for the *steerAmount* variable: one for half of the steer (*low*), and another one for double of the steer (*high*) (See Table 5.4).

Table 5.4 Output variables of the cognitive collision avoidance

Variable	Description	Fuzzy sets
steerAmount	Refers to how much the steering wheel should be turned to avoid the collision.	Zero, low, medium, high.
gasAmount	Refers to how much the vehicle should brake or accelerate.	Brake high, brake low, accelerate low.

In Table 5.4. we also mention the output variable *gasAmount*, which controls the acceleration or deceleration of the vehicle; there are three fuzzy sets associated to this variable: "Brake high" to fully apply the brakes, "Brake low" to slightly apply the brakes, and "Accelerate low" to press the accelerator. Figures 5.8 and 5.9 depict the membership functions for the fuzzy sets of *steerAmount* and *gasAmount*, respectively.

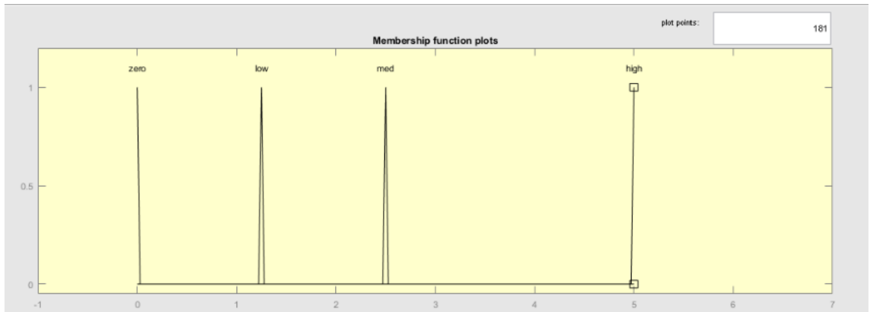

Figure 5.8 Membership function for *steerAmount*

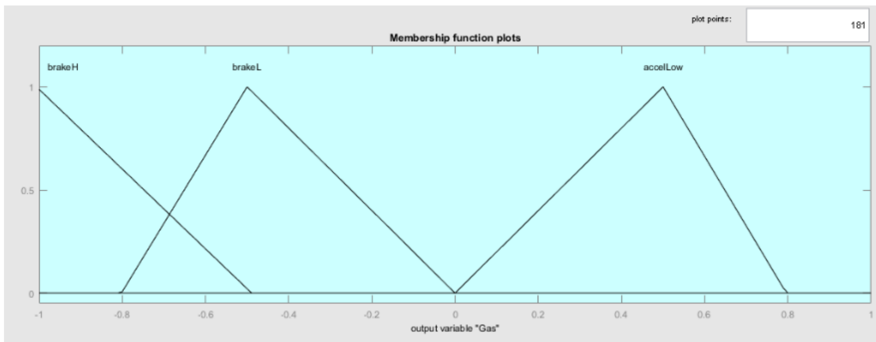

Figure 5.9 Membership function for *gasAmount*

After having defined the architecture of the collision avoidance system, including the details of the fuzzy controllers, in the next section we describe how it is integrated as the decision-making process of an intelligent agent that drives a vehicle in a cooperative environment.

5.6 Simulation of cooperative maneuvers

5.6.1 Agent architecture model

Prior to the design of the scenarios presented in chapter 2 as part of a simulation, integrating the solution elements explained in the previous section, we will begin by defining a common architecture model for the agents that will control the vehicles.

The BDI model, described in chapter 3, has a modularized approach that clearly divides responsibilities among its components; the perception, action and decision-making processes resemble the requirements of our cognitive agent. Considering our context awareness objective, we propose a simpler architecture, with fewer steps, and receiving as input the ego-sensors' data and also the intention information coming from neighbouring vehicles. The knowledge base is integrated to the architecture as a main component, continuously updated and in communication with the input and decision-making processes. Figure 5.10 depicts this proposal.

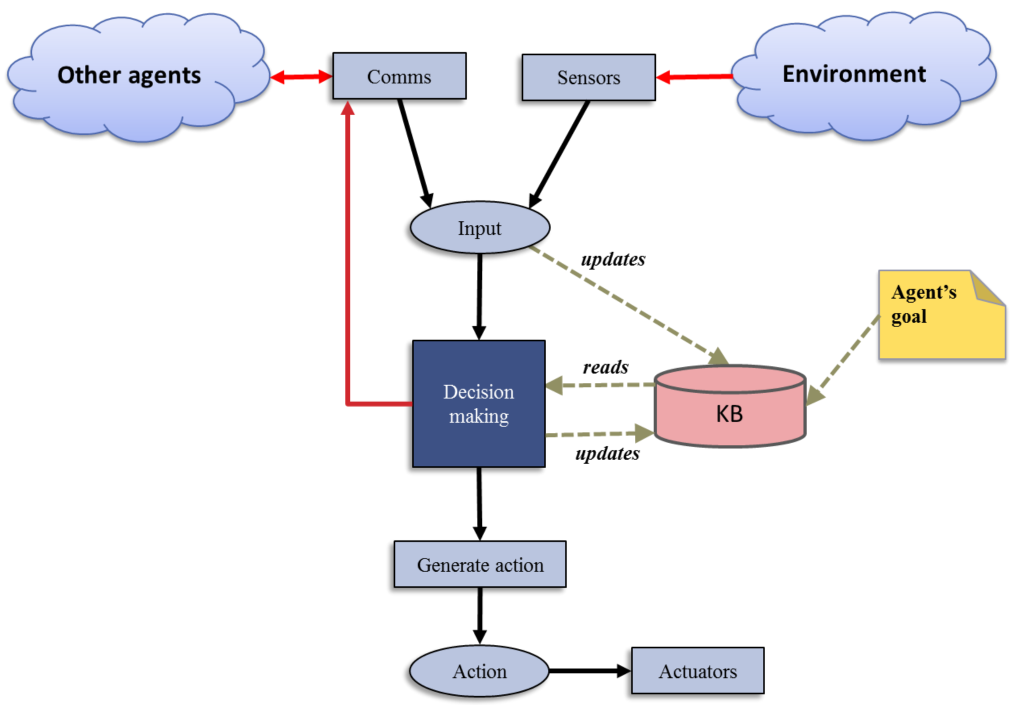

Figure 5.10 Proposed agent architecture

The logic behind this structure begins by the agent perceiving its environment via its on-board sensors; one of the novelties of the approach is the integration of the intentions of neighbouring vehicles to the agent's input. Then, both parts of the input are stored in its knowledge base as the vehicle's context. Later, given its goal and current context, the output of the decision-making process is produced in the form of an intention. The intention and sensor data are sent to vehicles in the vicinity, so they can integrate it to their own knowledge bases. Finally, the intention can be translated into actuators actions, i.e. acceleration/braking and steering wheel maneuvers. Permanently, the agent is perceiving the new state of the environment and updating its knowledge about what is going on; which can include: road state and neighboring vehicles' information.

5.6.2 Agent's logic flow

We consider in this research that vehicles have, mainly, two behaviors: driving and avoiding collision. We will integrate the works of [66] and [123] for trajectory planning and driving control, respectively; as discussed in chapter 3, the work of [66] is the basis of relevant works in the research of autonomous vehicles. Then, we propose to extend them by incorporating cooperative collision avoidance capabilities, based on context awareness.

The trajectory planning algorithm of [123] receives a start and goal point in a grid environment of an autonomous vehicle. After defining non-allowed zones in the grid, the proposal implements a modified version of the A* algorithm that finds the optimal sequence of Reeds&Shepp curves comprising the wheel rotations needed to get the vehicle to its destination. A planned trajectory is produced in the form of a set of points in the grid, and an expected velocity in each one of them; the autonomous car must follow it to arrive to the goal point.

Consequently, an algorithm to follow the planned trajectory is required to drive the vehicle and keep it on the track while it is not avoiding a collision. With this purpose, we will integrate to the solution the formulas proposed by Thrun et al. [66] for trajectory following. As was explained in section 3.2.1, the formulas in [66] produce the necessary speed and heading angle to keep the vehicle on track. However, the agent in control of the vehicle uses the actuators (pedals and steering wheel) to drive at an expected speed and to maintain an expected heading angle. Therefore, we need to compute which is the force required on the pedals in order to get the vehicle to the expected speed. Based on the work of Genta [124], in Appendix C we present the required formulas to calculate this force.

To properly simulate the agent's execution, we integrate the trajectory planning and trajectory following algorithms as part of the agent's logic (Figure 5.11). Our proposed cognitive processes, for collision detection and avoidance, are also integrated to this scheme; they decide what the agent should do next, given current knowledge of the situation. The agent will have two main behaviors: driving on the planned trajectory, and avoiding collision.

The architecture of the proposed avoidance module, depicted in Figure 5.7, constitutes the "Cognitive collision avoidance" block of the agent's logic flow presented in Figure 5.11.

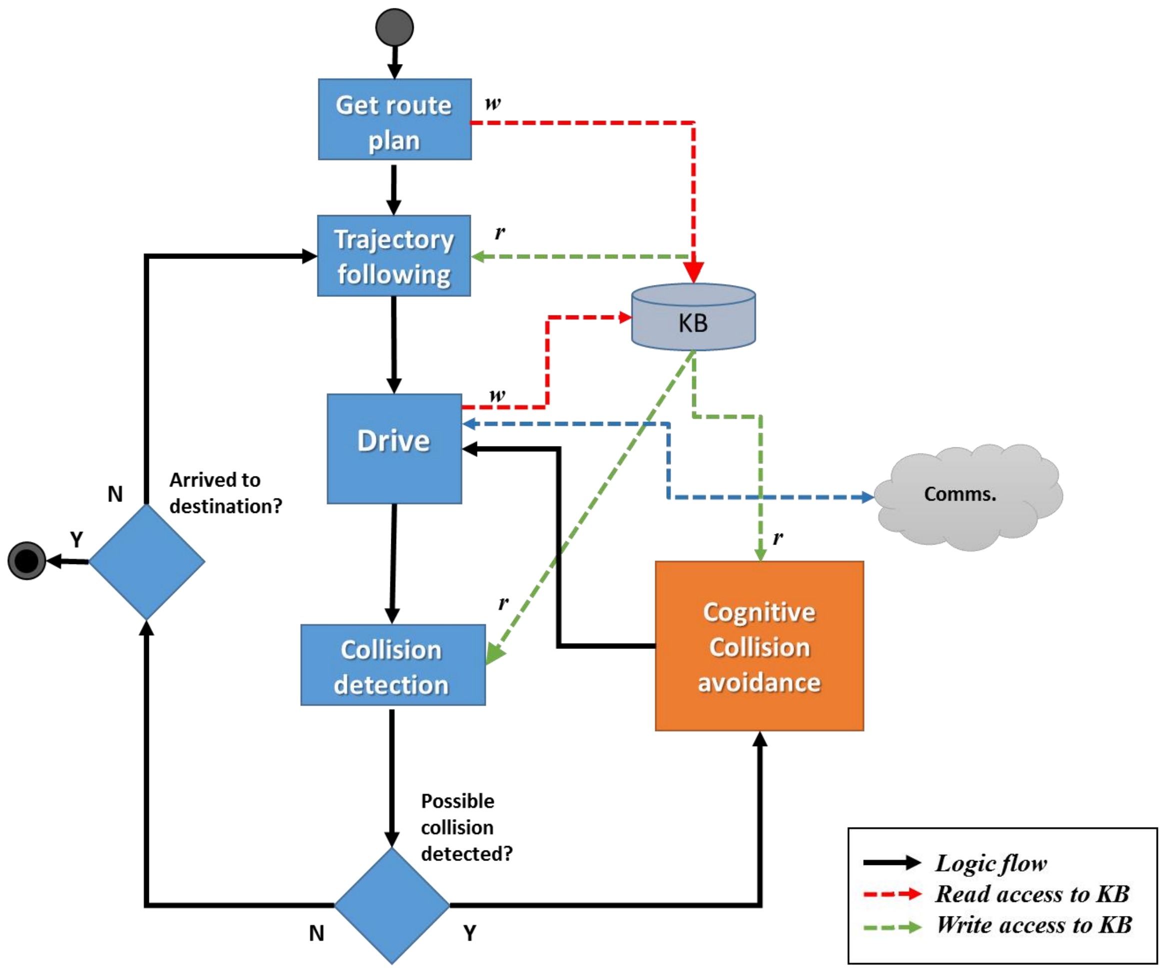

Figure 5.11 Logic flow of the agent driving a vehicle

The Algorithm 5.1 is a pseudocode representation of the agent's logic presented in Figure 5.11.

Algorithm 5.1. Pseudocode of the agent's logic

```
COMPUTE ROUTE_PLAN
GET ACCEL, STEER FROM trajectoryFollowing
EXECUTE ACCEL, STEER IN ACTUATORS
REPEAT
  CALL collisionDetection
  IF(collision_detected) THEN
    GET ACCEL, STEER FROM cognitiveAvoid
  ELSE
    GET ACCEL, STEER FROM trajectoryFollowing
  END_IF
  EXECUTE ACCEL, STEER IN ACTUATORS
  broadcast(INTENTION, SENSOR_INFO)
```

```
UNTIL arrivedDestination
```

As can be observed in Algorithm 5.1, the main goal of the vehicle agent is to arrive to its destination; however, this goal is subject to a constraint of collision avoidance. So, if the agent does not detect a dangerous situation in its immediate future, it will continue with the route plan. As explained in the previous section, the *trajectoryFollowing* algorithm generates the required acceleration and steering wheel values to drive the vehicle over the planned trajectory. The *intention* of the vehicle indicates its intended direction (left, right or continue straight) when avoiding a collision situation. The *broadcast* function sends the intention to other vehicles in the vicinity, so they can incorporate this information to their own knowledge base.

As discussed in the previous sections, the *collisionDetection* function allows the vehicle to detect if a collision situation might take place, and the *cognitiveAvoid* function triggers the collision avoidance fuzzy system in order to generate the necessary acceleration and steering actions.

To validate our proposed solution, comprised by the agent architecture of Figure 5.10 and the logic flow depicted in Figure 5.11, we must create collision scenarios and test how the vehicles behave in such situations. However, testing with real vehicles involves a huge risk and costs that are prohibitive for this research; so, we aim to assess the validity of our proposal by means of a simulation environment. In the following section, we explain the development of such environment and the integration of our collision avoidance system.

5.6.3 Simulation environment

In order to test the agents and properly validate our approach, we have designed and implemented a 3D visualization tool that will allow us to model the vehicle and the environment to simulate the problem scenarios. As illustrated in Figure 5.12, we need to integrate the agent's cognitive model to the visualization tool; by doing so, we will be able to design the collision scenarios in a 3D environment and observe how the simulated vehicles execute cooperative maneuvers.

As result, we will have vehicles that produce cooperative avoidance maneuvers that effectively and timely avoid the dangerous situations conceived in the proposed scenarios. In this section, we explain the requirements of the 3D visualization tool and the details of its implementation.

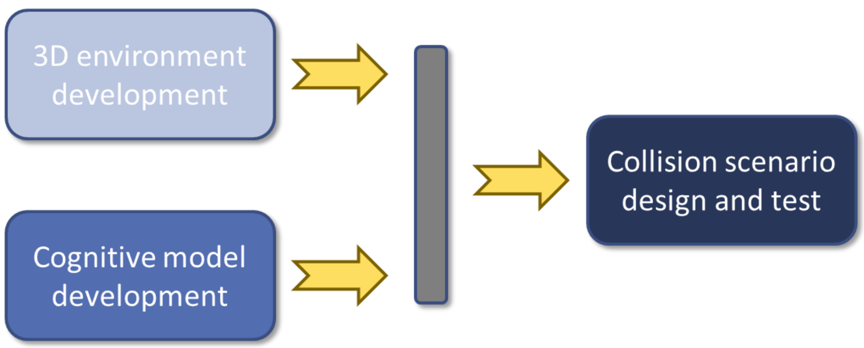

Figure 5.12 Phases of the validation process using 3D visualization tool

The 3D simulation environment was built using the XNA graphic engine; which allows the researcher to observe, organize and move objects in a three-dimensional space. This environment is configured to have a refresh rate of 10Hz; such rate applies for updating the vehicles' controls, sensor values and messages. The *Farseer* function library is a physics engine containing classes and methods to simulate the movement of objects in a 3D environment, it was integrated to the solution in order to provide realistic behavior to the simulated vehicles while they move. This engine controls and simulates the behavior of rigid bodies when they are affected by, external or internal, physics forces such as: acceleration, gravity and collisions. It applies the set of forces over every object, considering its location, mass and other physic attributes to compute its acceleration, its velocity and therefore its location in the 3D space. The main class used in the physics engine is *3DBody*, every simulated object in the environment must inherit from it; Figure 5.13 depicts a diagram relating the classes of the simulation objects in our work with the physics engine.

Complete detail on the description of objects, as well as the variables and constraints that can be set for vehicles in the simulation, can be found in Appendix B.

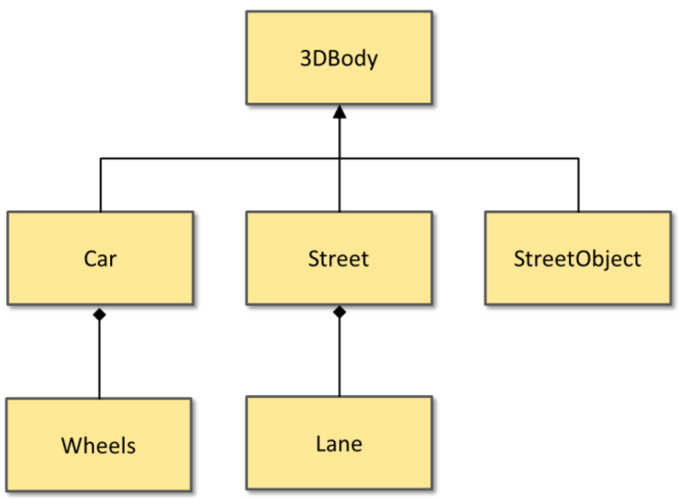

Figure 5.13 Class diagram of 3D simulated objects

As the physics engine uses forces to compute acceleration, velocity and location of the objects in the map, the driving control must compute the required force (positive or negative) to approximate the vehicle to the path.

In some cases, it will be necessary that vehicles generate and execute avoidance maneuvers even in the presence of other non-connected vehicles. To simulate non-connected and non-autonomous vehicles we need a traffic simulator that produces realistic driving behavior for them; such program can control the general aspects of the simulations, like: deciding when a new vehicle enters the simulation, where that vehicle goes, its speed, when it exits, etc.

SUMO (Simulation of Urban MObility) [125] is an open source microscopic traffic simulator that can model large vehicular networks; it provides the possibility to configure user defined road networks, or load them from OpenStreetMap format files. An advantage of this simulator is that it provides an API (Application Programming Interface) to remotely control the simulation; it is relevant because with it we can connect our 3D environment and obtain the traffic status as it changes in real time. A comparison of the capabilities of this and other simulators was made by the author in [126].

Therefore, in scenarios with multiple types of vehicles, SUMO will be controlling the non-autonomous vehicles and the agents in our 3D visualization tool will be in charge of driving the autonomous cars with collision avoidance capabilities. As both types of vehicles will be at the same time in the simulations, we need to model a structure that integrates our 3D visualization environment with the traffic simulator; with this in mind, we propose the class diagram depicted in Figure 5.14.

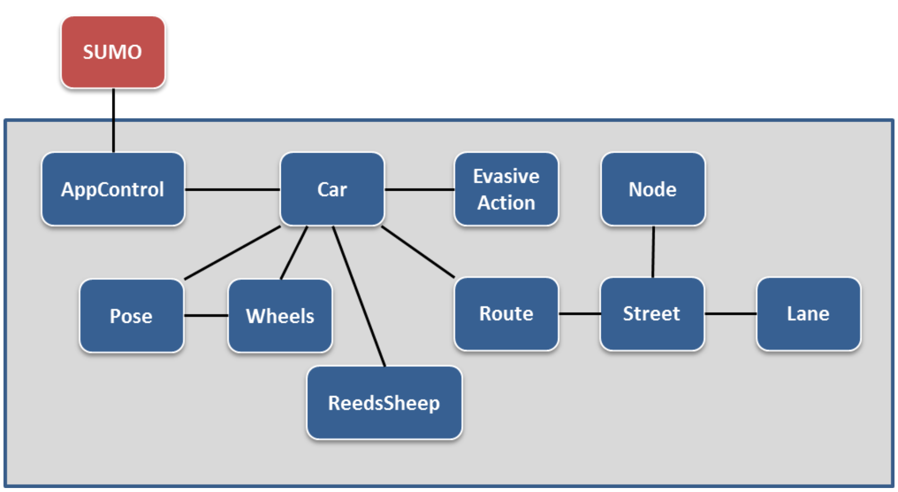

Figure 5.14 Proposed class diagram for the integration model

The simulation scenario is loaded, in parallel, in the traffic simulator and into the local classes schema of our proposed environment, in order to create the three-dimensional representation. The integration is achieved via online interaction using the Traffic Control Interface (TraCI) [125], which is provided as one of the modules of SUMO. This interface has a client-server architecture based on TCP to allow online access to an ongoing traffic simulation. We propose to use a single set of configuration files, shared between both programs to guarantee that they run the same scenario; Figure 5.15 shows graphically the proposed integration between the 3D environment and the traffic simulator.

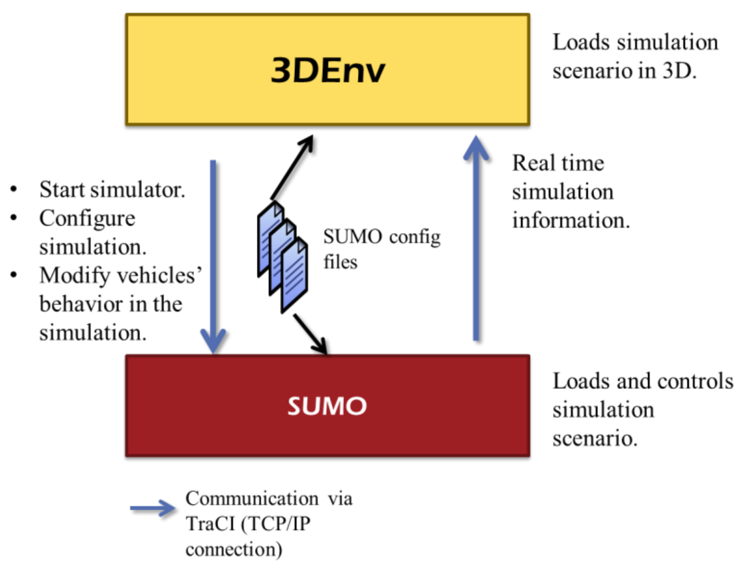

Figure 5.15 Interaction between project components

Chapter VI
Generalization of the avoidance system

6.1 Introduction

In chapter 5 we described our proposed collision avoidance architecture, which is composed by two levels: 1) a fuzzy system that uses context information to decide which is the safest direction the autonomous vehicle can take in order to avoid a collision. And 2) a fuzzy controller that generates the steer and brake (or accelerator) controls required to safely drive the vehicle to the target lane. So, considering data from vehicles in the vicinity, the first controller provides context awareness to the autonomous car; this is done through the linguistic rules, which produce a conclusion that depends on the values of the linguistic variables described in section 5.5.3.

The set of rules of the first fuzzy system was empirically conceived, using intuition for a group of possible situations that can be represented with the linguistic variables; essentially, the goal was to make tests to verify the correct operation of the proposal under controlled situations. However, as it will be exposed in this chapter, there is a large number of situations that can be represented with the proposed approach, and ideally the collision avoidance controller must be able to drive an autonomous vehicle away from the danger in any case. Therefore, our objective is to generalize this approach, by generating a complete set of rules that is able to correctly guide the vehicle in a wider range of situations. Our proposal in this chapter is based on the formalization of the problem of finding a general set of rules, and presenting a solution for it.

Our generalization approach is presented in this chapter as follows: first, in section 6.2 we make an analysis of the problem. Later, in section 6.3, we explain the parameters of the problem; and, in section 6.4 we present our proposal for solution. In section 6.5 we explain our proposal of a search algorithm for the rules, as well as an analysis of the worst-case scenarios. Finally, in section 6.6 we describe the experiments and discuss the results.

6.2 Analysis of the generalization problem

We will start by explaining how the environment of the vehicle is related to the inputs of our proposed system. The vehicle context, as presented in chapter 5, is replicated here in Figure 6.1; also, in the remaining of this chapter, the context is related to the vehicle marked as "*v0*", which is the one with collision avoidance capabilities.

For each of the six distance and speed variables taken from the vehicle context, and used as input for the first fuzzy system in the architecture, we have identified 4 possible sets of values (*linguistic values*); the descriptions of these variables and the values of the corresponding sets are replicated in Table 6.1.

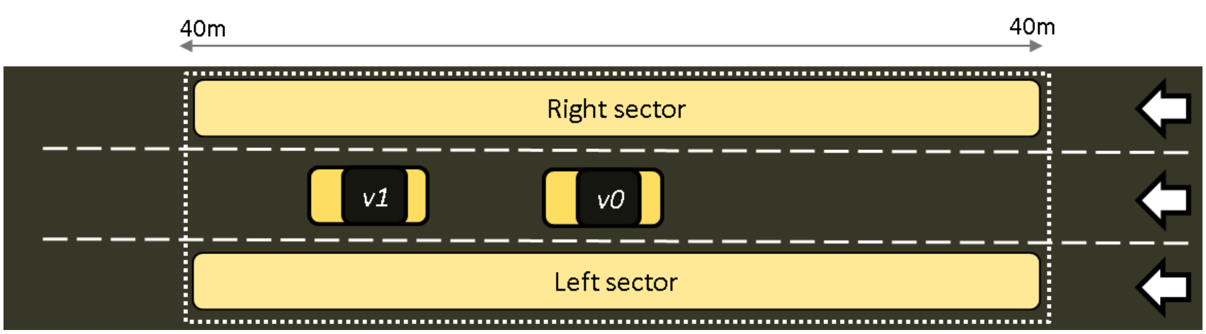

Figure 6.1 Vehicle context in the highway environment

Table 6.1 Input and output variables of the first stage of the avoidance system

Variable	**Type**	**Description**	**Possible values**
distLeft	Input	The distance of *v0* to the nearest vehicle on the left lane.	Far, medium, close and too close.
deltaLeft	Input	The difference in the speed of *v0* relative to the closest vehicle on the left lane.	Fast, regular, slow, away.
distFront	Input	The distance of *v0* to the vehicle in front.	Far, medium, close and too close.
deltaFront	Input	The difference in the speed of *v0* relative to the vehicle in front.	Fast, regular, slow, away.
distRight	Input	The distance of *v0* to the closest vehicle on the right lane.	Far, medium, close and too close.
deltaRight	Input	The difference in the speed of *v0* relative to the closest vehicle on the right lane.	Fast, regular, slow, away.
direction	Output	Direction of the avoidance maneuver	Left, Center, Right.

For each combination of values of the input linguistic variables must exist a rule; this rule produces a direction (left, center or right), which we can observe as a classification of the context. Thus, the antecedent of the rule is the context representation and its conclusion can be seen as a classification, which we use as an indication of the direction of the avoidance maneuver.

Accordingly, the set of all the rules of the fuzzy system corresponds to the avoidance strategy for the contexts that can be represented by the defined input variables. Given that there are four possible values for each of the six input variables (View Table 6.1), we can represent at most $4^6 =$ 4.096 collision contexts. And for each one of them we must define a rule that indicates the avoidance direction. This set of 4.096 rules enables the fuzzy system with an answer for any possible combination of input variables.

Moreover, considering the highway environment defined for this research (Figure 6.1), our problem is to find, for every rule, the conclusion that minimizes the risk of collision. In the following sections, we explain how we quantify the value of a set of rules to ultimately find the optimal one.

6.3 Problem parameters

6.3.1 Linguistic rules

By combining the possible values of the variables described in Table 6.1, and using the format of linguistic rule presented in section 5.5.3, we can write the rules of our proposed system; for instance, a sample rule can be written as:

```
IF distLeft IS Far AND deltaLeft IS High AND distCenter IS Far AND deltaCenter IS
High AND distRight IS Far AND deltaRight IS High THEN Direction IS Center
```

Following the same pattern, we can write the antecedents of the 4.096 rules of our system; and, as mentioned in the last section, our goal is to find the consequents for those rules. Therefore, we need to define a quantitative way to compare two of them and indicate which one is *better* than the other. Given that we will use these rules as the core of the fuzzy system that minimizes the risk of collision, then a consequent will be *better* than other if it produces an inferior risk of collision for a given antecedent.

6.3.2 Quantification of risk of collision

From section 5.5.3, we know that the output of the fuzzy system in the first stage determines which controller is called in the second stage. Therefore, the performance of the system as a whole depends on the synergy of systems in both stages. The result of the execution of the maneuver in the second stage will be an indicator of how good the decision of the first system was. And therefore, how good the rule is for that particular context; e.g. if the second stage system produces a collision-free maneuver, we can say that the avoidance direction indicated by the first system was correct. Moreover, knowing that distance is more important than comfort when a vehicle faces a collision situation, we have implemented a multi-objective function that weighs these two variables with a domination criterion to compute the quantitative value of a particular rule; in the rest of this section, we explain the details of such function.

In regards to the comfortability, Naranjo et al. [107] explain that the smoothness of a maneuver is directly related to the oscillations in steer and acceleration. Thus, we have defined equation (6.1) to consider the oscillations in the maneuver.

$$manOsc = Osc_{ac} * \omega_{ac} + Osc_{st} * \omega_{st} \tag{6.1}$$

Where, Osc_{ac} and Osc_{st} are normalized values that indicate the acceleration and steering changes in the maneuver, respectively. A value of 0 in these variables means there were no oscillations during the maneuver, and a value of 1 indicates the presence of oscillations during the whole maneuver. ω_{ac} and ω_{st} are coefficients that weigh the importance of the acceleration and steering oscillations, respectively.

To keep *manOsc* in the interval [0, 1], the sum of the weights must be equal to 1 (6.2).

$$\omega_{ac} + \omega_{st} = 1 \quad (6.2)$$

So, for instance, if we set ω_{ac} = 0.8 and ω_{st} = 0.2, then *manOsc* will be biased towards the oscillations in the acceleration. On the contrary, if we set ω_{ac} = 0.2 and ω_{st} = 0.8, then *manOsc* will be biased towards the oscillations in the steer. For the purposes of this research, we consider that both variables are equally important to compute the degree of oscillation in the maneuver; therefore, ω_{ac} = 0.5 and ω_{st} = 0.5.

Consequently, the smoothness of the maneuver can be defined as the opposite of its oscillation:

$$manSmooth = 1 - manOsc \quad (6.3)$$

Taking in consideration the vehicle with collision capabilities (*v0* from Figure 6.1), it is clear that there can be other cars in its environment. And given that the goal is to avoid the collision, we will only consider the distance to the closest one. Therefore, equation (6.4) defines *minDist* as the minimal distance of *v0* with respect to all others vehicles in its context.

$$minDist_{v0} = min(dist(v0, j)), \forall j \in Ct_{v0}, v0 \neq j \quad (6.4)$$

Where Ct_{v0} denotes the group of vehicles that are in the context of *v0*.

Then, we aim to find the rule set *S* that produces an avoidance maneuver with minimal oscillations and which maximizes $minDist_{v0}$ while avoiding a collision. In order to combine the distance and the oscillations in a unique multi-objective function, we limit $minDist_{v0}$ to the interval [0, *safeDist*] and normalize using equation (6.5).

$$minDistNorm_{v0} = \begin{cases} \dfrac{minDist_{v0}}{safeDist}, & if\ minDist_{v0} < safeDist \\ 1, & if\ minDist_{v0} \geq safeDist \end{cases} \quad (6.5)$$

Since *manSmooth* is already in the interval [0, 1], the result of a maneuver performed by can be defined as:

$$M_{v0} = minDistNorm_{v0} * \ \omega_{dis} + manSmooth_{v0} * \omega_{sm} \quad (6.6)$$

Where, *v0* is the vehicle with collision avoidance capabilities, $minDistNorm_{v0}$ and $manSmooth_{v0}$ are the normalized minimal distance and the smoothness of the maneuver, respectively. ω_{dis} and ω_{sm} are coefficients that weigh the importance of the minimal distance and the smoothness of the maneuver, respectively.

To keep M_{v0} normalized in the interval [0, 1], the sum of the weights must be equal to 1:

$$\omega_{dis} + \omega_{sm} = 1 \quad (6.7)$$

To explain the effect of the weights on the maneuver value, we can consider for instance setting $\omega_{dis} = 0.7$ and $\omega_{sm} = 0.3$, which produces M_{v0} to be biased towards the minimal distance between vehicles and give less importance to the comfortability of the occupants. On the contrary, if we set $\omega_{dis} = 0.3$ and $\omega_{sm} = 0.7$, then M_{v0} will be biased towards the smoothness of the maneuver. For the purposes of this research, we consider that keeping the vehicle away from others is more important than the comfort of the avoidance; therefore, we use $\omega_{dis} = 0.7$ and $\omega_{sm} = 0.3$.

Thus, we must find the optimal rule conclusions; to achieve this goal we must explore the solution space and select the conclusion that maximizes M_{v0} for each of the 4.096 rules. In the next section, we make a reasoning that allows us to decide which solution strategy is the most appropriate.

6.4 Solution strategy

According to the analysis made in the previous sections, we have 4.096 contexts that are represented by a rule, using the proposed variables; each rule has three possible conclusions (left, center, right); then, we must find the conclusion that maximizes M_{v0} for each given context/rule. Such description can be seen as a combinatorial optimization problem.

Two possible practices can be considered to solve this type of problems: metaheuristics and exact methods. Metaheuristics are known for solving complex optimization problems [127], especially those where the time required to explore the whole search space is prohibitive; they usually follow a gradient minimization pattern that rapidly leads them to a local-optima result. Random parameters avoid these algorithms from falling into poor local-optima and jump to a different zone of the search space to find better solutions.

On the other hand, exact methods make an exhaustive search to evaluate every possible solution in order to find the global-optima [127]; depending on the size of the search space, these methods might take a long time to converge. Therefore, an assessment of search-time vs solution-quality must be considered to decide which solution strategy best fits the problem in place.

It is worth noting that finding the conclusions for the rules is an offline process that will occur only once, before the system can be tested and used. Thus, since a real-time response is not required, at this point we can neglect the execution time in favor of the quality of the solution. Later, when the system is configured with the rules and their conclusions, it will be triggered online to produce immediate responses to a risk of collision situation.

As previously stated, in our particular problem, there are three possible values for each rule; thus, the size of the search space is 4.096 * 3 = 12.288 solutions. We know that triggering the fuzzy system and obtaining an output for a given input takes less than 0.1s; thus, trying all the 12.288 possibilities would take approximately 20 minutes (1.288,8s). For the purposes of this research, this can be considered as a reasonable execution time; moreover, given that a sub-optimal solution is not desirable in a collision avoidance context, we have opted to implement an exhaustive search algorithm. We describe the details of such implementation in the next section.

6.5 Search algorithm

6.5.1 Rule representation in code

In the previous section we stablished equation (6.6) as a quantitative value for an evasive maneuver; where higher values represent those maneuvers that keep *v0* away from other vehicles. Also, according to the architecture presented in Figure 5.7, the execution of one avoidance maneuver or another depends on the decision made by the fuzzy system on the first stage, which is controlled by its internal set of rules. With this in mind, in this section, we define an algorithm that finds the conclusions that produce the maximal value of M_{v0} for each rule.

In order to build the search algorithm, we start by writing the antecedents of the rules; to do so, we have conceived the function *generateRules*, which creates the antecedents by combining the possible values of the linguistic input variables.

To avoid using symbolic values in our algorithm, we opted for a numeric representation, based on the code presented in Table 6.2, for the values of the input variables (Table 6.1). Similarly, the three possible values of the output variable: *left*, *center* and *right*, are also codified as numbers; this correspondence is presented in Table 6.3.

Table 6.2 Correspondence of values of input variables to codified values

Variable	Original value	Codified value
distLeft, distFront and distRight	Far	1
distLeft, distFront and distRight	Medium	2
distLeft, distFront and distRight	Near	3
distLeft, distFront and distRight	Too near	4
deltaLeft, deltaFront and deltaRight	High	1
deltaLeft, deltaFront and deltaRight	Medium	2
deltaLeft, deltaFront and deltaRight	Low	3
deltaLeft, deltaFront and deltaRight	Away	4

Table 6.3 Correspondence of values of the output variable *direction* to codified values

Original value	Codified value
Left	1
Center	2
Right	3

To store the codified rules, the *generateRules* function uses an array of size 7, identified as *Rule*, where the first six elements represent the input linguistic variables of a rule, and the last element represents the conclusion of the rule. We present in Table 6.4 the correspondence of items in the *Rule* array to the linguistic variables.

Table 6.4 Correspondence of variables to items of the *Rule* array

Item in *Rule* array	Variable
1	distLeft
2	deltaLeft
3	distFront
4	deltaFront
5	distRight
6	deltaRight
7	direction

We depict the pseudocode of function *generateRules* in Algorithm 6.1. This function receives a parameter (*item*) that indicates which index of the rule array is being set by the function (Line 3 of Algorithm 6.1); when the *item* index is in the interval [1, 6], it means that the function is setting the value of an input variable in the rule (Lines 8-10 of Algorithm 6.1). Otherwise, if the *item* index is 7, it means that the function should set the conclusion of the rule, which at this point we set as 2 (*center*) by default (Line 5 of Algorithm 6.1).

Algorithm 6.1. Pseudocode of the *generateRules* function

```
1   Rules = List   //List to store all the rules
2   Initialize Rule to [1, 1, 1, 1, 1, 1, 1] //Array to store a single rule
3   generateRules (item)  //Beginning of the function generateRules
4    IF item = 7 THEN  //Set the conclusion?
5     Rule[7] = 2  //Set the conclusion to Center as default
6     Rules.insert(Rule) //Insert the new rule in the list of rules
7    ELSE
8     REPEAT FOR Val = 1 TO 4 //Loop for the 4 possible values of an input variable
9      Rule[item] = Val  //Set the variable value in the rule array
10     generateRules(item + 1)  //Call the function to set the value of next variable
11    END REPEAT
12   END IF
```

Algorithm 6.1 ends when all possible combinations of values for the input variables have been generated in the form of coded rules and stored in the ***Rules*** list. Next step is to find the conclusions for all the rules in this list. We describe this process in the next part.

6.5.2 Algorithm to find rules' conclusions

Using the rule representation described in section 6.5.1, and the ***Rules*** list generated by Algorithm 6.1, we have created a function that finds the conclusion for each rule. It makes an exhaustive search for these conclusions by trying with each possible value (left, center, right) and saving the one with higher M_{v0}. We depict this function in Algorithm 6.2.

Algorithm 6.2. Pseudocode of the exhaustive search for rules' conclusions

```
1   Rules = generateRules(1)  //Get the rules
2   N = 4096  //Number of rules
3   REPEAT FOR I = 1 TO N  //Loop over all the rules
4     bestMvo = 0  //Stores the best Mvo for the current rule
5     bestDir = 1  //Stores the direction corresponding to the best Mvo
6     REPEAT FOR dir = 1 TO 3  //Loop over the 3 possible avoidance directions
7       Rules[I][7] = dir //Set the conclusion to the testing value on the rule I
8       Mvo = getManeuverValue(Rules[I]) //Get the Mvo when using this
                                           conclusion (avoidance direction)
9       IF Mvo > bestMvo THEN //If the Mvo is higher than current best then
                                replace current best with new Mvo
10        bestMvo = Mvo  //Save new best Mvo
11        bestDir = dir  //Save the direction associated to the best Mvo
12      ENDIF
13    END REPEAT
14    Rules[I][7] = bestDir  //Set the conclusion of the rule as the direction of
                               the highest Mvo for rule I
15  END REPEAT
```

In Algorithm 6.1, we defined ***Rules*** as a list of arrays, where each array represents a rule; then, in Algorithm 6.2, we iterate by every rule in this list (Line 3 of Algorithm 6.2) and evaluate the value of M_{v0} three times, one for each possible value of the conclusion (Lines 6-13 of Algorithm 6.2). To compute M_{v0}, the function *getManeuverValue* (Line 8 of Algorithm 6.2) uses the conclusion of the received rule to select the appropriate fuzzy system of the second stage, and triggers it using the context configuration found in the antecedent of the rule. For each rule in the list, we save the direction that produced the highest M_{v0} (Lines 9-12 of Algorithm 6.2), and use it as its definitive conclusion (Line 14 of Algorithm 6.2).

6.5.3 Analysis of worst case scenarios

Starting on the basis of the environment presented in Figure 6.1 and the context variables defined in the previous chapter, in this section we make an analysis of the worst possible cases that can trigger our collision avoidance system. We will also demonstrate that our proposal will guarantee the collision avoidance associated to these worst-case scenarios.

To create the collision risk for “*v0*” we will set “*v1*” facing an unforeseen event that forces it to fully apply the brakes. In consequence, “*v0*” must trigger the avoidance system in order to avoid such collision. We consider the fact that with fewer avoidance possibilities, the vehicle involved in the collision risk is in a higher danger; thus, in this analysis, in order to create the worst-case scenario, we will arrange the environment with “*v0*” surrounded by other vehicles on both contiguous lanes of the street depicted in Figure 6.1. Similarly, since we will analyze how the avoidance system on “*v0*” reacts to “*v1*” fully braking, because of the unexpected event, the scenario must start in a no-collision risk state.

As we explained in section 5.5.2, the avoidance system is triggered when the TTC is less than 2 seconds or when the distance is shorter than the safe distance *Ds* (6m). Since we have two variables that can trigger the system, in this section we will analyze the two corresponding worst-case scenarios: one associated to the minimal inter-vehicular distance, and the other associated to the TTC.

6.5.3.1 Minimal inter-vehicular distance scenario analysis

In this case we will determine the configuration of the worst-case scenario associated to the inter-vehicular distance; later, we will analyze how the collision avoidance system performs in such scenario when the inter-vehicular distance drops below the minimal, i.e. when "*v0*" and "*v1*" are separated by less than 6m. As we assume that at the start there is no collision risk, we locate both vehicles on the scenario separated by 6m. The next step is setting the vehicles speeds.

When configuring the vehicles speeds, we have three options: "*v0*" moves faster than "*v1*", "*v0*" moves slower than "*v1*", and "*v0*" moves at the same speed as "*v1*". If we set "*v0*" moving faster than "*v1*" then the inter-vehicular distance will be less than 6m, which represents a collision risk before the appearance of the unforeseen event; this situation contradicts our assumption of no-collision risk at the start. If we set "*v0*" going slower than "*v1*" then the inter-vehicular distance is higher than 6m, which no longer represents the worst-case scenario. Therefore, for our analysis, the only option is setting both vehicles with the same speed; this fulfils our assumption of a worst-case scenario starting with no-collision risk. For the purposes of this demonstration, we set the speeds at 60kph.

At the start (*t0*), since both vehicles go at the same speed, they keep the 6m distance, thus there isn't any risk of collision. However, when an unforeseen event (*E*) occurs, "*v1*" is forced to apply the brakes; taking into account the study of Lee et al. [104], we consider that the acceleration of "*v1*" when it is braking is of $-3m/s^2$.

To compute the TTC we use equation (5.4), which requires that speeds are introduced in meters per second (mps); thus, for the case we are studying, the speed for both vehicles is 16.662mps. As mentioned in section 5.6.3, the refresh frequency of the sensors is assumed to be of 10 times per second; then, we can use the physics formula for speed (6.8) to compute the speed of the vehicle after each timestep.

$$v_f = v_0 + at \tag{6.8}$$

Where v_0 is the initial speed, a is the acceleration, t is the elapsed time, and v_f is the final speed after the elapsed time. Therefore, as "*v1*" brakes, its speed after 0.1 seconds of the event E can be computed as:

$$v_f = 16.662mps - 3\,{}^{m}/_{s^2} \cdot 0.1s$$

So, the speed of "*v1*" is reduced to 16.362mps. And with this new speed, the value of equation (5.4) for TTC is now:

$$TTC = \frac{D_{act}}{V_0 - V_1} = \frac{6}{16.662 - 16.362} = 19.9s$$

To give an idea of the importance of considering both the distance and the TTC to assess the collision risk, we present in Table 6.5 the evolution of speed, distance and the TTC of "*v0*" and "*v1*", when "*v1*" has an acceleration of -3m/s^2.

Table 6.5 Change of speed, distance and TTC in first worst-case scenario

	Time (secs)	speed v0 (mps)	speed v1 (mps)	accel v0 (m/s^2)	accel v1 (m/s^2)	Distance (m)	TTC (secs)
t0	0	16.662	16.662	0	-3	6	
t1	0.1	16.662	16.362	0	-3	5.97	19.9
t2	0.2	16.662	16.062	0	-3	5.91	9.85
t3	0.3	16.662	15.762	0	-3	5.82	6.4667
t4	0.4	16.662	15.462	0	-3	5.7	4.75
t5	0.5	16.662	15.162	0	-3	5.55	3.7
t6	0.6	16.662	14.862	0	-3	5.37	2.9833
t7	0.7	16.662	14.562	0	-3	5.16	2.4571
t8	0.8	16.662	14.262	0	-3	4.92	2.05

t9	0.9	16.662	13.962	0	-3	4.65	1.7222
t10	1	16.662	13.662	0	-3	4.35	1.45

As can be observed in Table 6.5 at *t1* (0.1s), even if the TTC is higher than 2s, the distance is already less than 6 meters. Therefore, the collision risk is detected and the avoidance system is triggered. It is only after 0.9s that the TTC is lower than 2s, and at that moment, the inter-vehicular distance is 4.65m. Thus, incorporating the distance as a component for the collision risk assessment has allowed us to trigger the avoidance system 0.8s before than just using the TTC; at current speed, this represents 1.32 meters of extra space to "*v0*" to execute its avoidance maneuver.

Since the unforeseen event forces "*v1*" to make a full stop, then "*v0*" must also brake to a full stop. According to Lee et al. [104], the braking distance can be computed as:

$$Dist_{brake} = \frac{V^2}{170} \tag{6.9}$$

Where *V* is the vehicle's speed in kph, and 170 is a constant. Thus, in our case, the braking distance of "*v0*" and "*v1*" are 21.18 and 20.42 meters, respectively. Since they had an initial distance of 6 meters then, at the end of the maneuver, after moving 21.18m and 20.42m, they will be separated by 5.24 meters; we depict this graphically in Figure 6.2. Our considerations, supported by calculations and equations from the literature, clearly demonstrate that our hypothesis on collision avoidance in the worst case associated to the distance is valid.

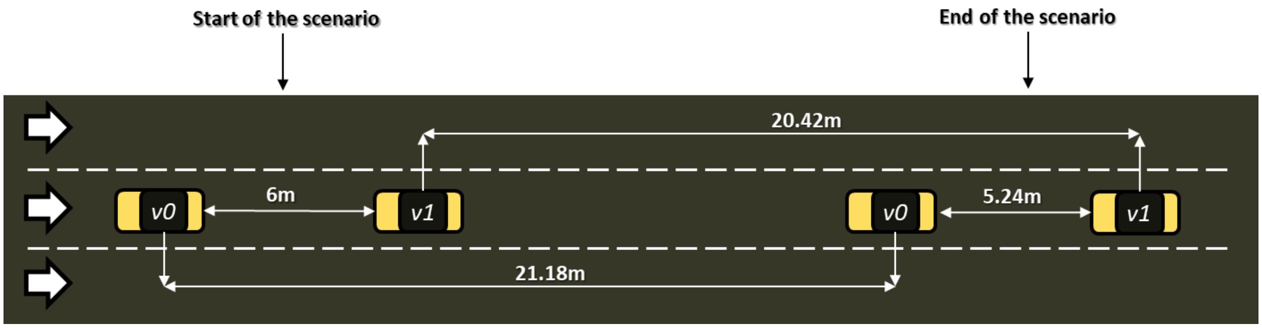

Figure 6.2 Distance-related worst-case scenario analysis

6.5.3.2 TTC triggered scenario analysis

In this part, we will determine the configuration of the worst-case scenario that triggers the avoidance system because of a TTC below the limit; afterwards, we will analyze how the collision avoidance system performs in such scenario when the TTC descends below 2 seconds.

We consider the environment as in the previous case, "*v0*" and "*v1*" on the same lane, other vehicles surrounding "*v0*" impeding a lane change maneuver, and we force "*v1*" to fully apply the brakes in order to create a collision risk for "*v0*". As was shown before, if both vehicles move at the same speed, the TTC is kept over the threshold; however, if "*v0*" moves slower than "*v1*" then their inter-vehicular distance is increased, thus increasing the TTC. On the contrary, if "*v0*" moves faster than "*v1*", the vehicles are approaching, reducing their inter-vehicular distance, and therefore decreasing the TTC.

Since we aim to trigger the avoidance system by means of the TTC, then we will set "*v0*" with a higher speed than "*v1*". According to our assumptions, the speed limit in the scenario is 60kph, then we set this value as the speed of "*v0*".

As in the previous case, the scenario must start with no-collision risk, therefore we must set an initial speed for "*v1*" that does not contradict this assumption. So, if we reduce the speed of "*v1*" then we must increase the initial inter-vehicular distance of both vehicles in order to comply with the aforementioned assumption. By setting a speed of 40kph for "*v1*", we can consider values for the initial inter-vehicular distance and select the one that generates the worst possible case. In Table 6.6 we show a list of possible initial distances and their corresponding TTC values when "*v0*" and "*v1*" move at 60kph and 40kph, respectively.

Table 6.6 TTCs associated to the initial inter-vehicular distances for worst-case scenario #2

Distance (m)	TTC (secs)
15	2.4624
14	2.2915
13	2.1207
12	1.9499
11	1.7791

As explained, in order to start with no-collision risk, the TTC must be over the 2 seconds threshold; and, to create the worst-case scenario, we must use the shortest possible inter-vehicular distance, which according to Table 6.6 is 13 meters. In Table 6.7 we show how the speeds, distance and TTC change for this case.

Table 6.7 Change of speed, distance and TTC in worst-case scenario #2

	Time (secs)	speed v0 (mps)	speed v1 (mps)	accel v0 (m/s^2)	accel v1 (m/s^2)	Distance (m)	TTC (secs)
t0	0	16.662	11.108	0	-3	13	
t1	0.1	16.662	10.808	0	-3	12.4146	2.1207
t2	0.2	16.662	10.508	0	-3	11.7992	1.9173
t3	0.3	16.662	10.208	0	-3	11.1538	1.7282
t4	0.4	16.662	9.908	0	-3	10.4784	1.5514
t5	0.5	16.662	9.608	0	-3	9.773	1.3855
t6	0.6	16.662	9.308	0	-3	9.0376	1.2289
t7	0.7	16.662	9.008	0	-3	8.2722	1.0808
t8	0.8	16.662	8.708	0	-3	7.4768	0.9400

t9	0.9	16.662	8.408	0	-3	6.6514	0.8058
t10	1	16.662	8.108	0	-3	5.796	0.6776

As can be observed in Table 6.7 at *t2* (0.2s), the TTC is lower than 2s, then the avoidance system is triggered. Finally, we use equation (6.9) to compute the braking distance of "*v0*" and "*v1*", which produces 21.18 and 8.42 meters, respectively. Since they had an initial distance of 13 meters, at the end of the maneuver they will be separated by 0.24 meters. As in the previous case, our considerations, supported by calculations and equations from the literature, clearly demonstrate that our hypothesis on collision avoidance in the worst case associated to the TTC is valid.

6.5.3.3 Coverage of all configuration cases

In section 5.5.3 we explained that the vehicles to be considered in the collision avoidance context are those within a range of 40 meters, and we detailed how the membership functions for the distance variables cover this whole interval. Similarly, the relative speed (delta) between any two vehicles is considered to be of interest for the vehicle's context if it is within the interval [-5, 20] kph. Furthermore, considering again Figures 5.4 and 5.5, we observe that the fuzzy sets on the boundaries of the membership functions are open, meaning that any possible value in the interval [-∞, ∞] is guaranteed to be covered.

As can be noticed in Table 6.1, these variables for distance and relative speed are considered independently for the vehicles on the left and right lanes; moreover, they refer only to the vehicle that is closer to "*v0*" on the respective lane. Since in our assumptions we stablished that vehicles trust in the information sent by others, then the variables' values are always valid and they are never going to represent an impossible scenario or situation.

We have also stablished that the proposed input variables serve to represent the vehicle’s contexts. Also, we defined the fuzzy sets that cover all the possible values that can take these variables. Therefore, a single combination of fuzzy values for these fuzzy sets represent a unique vehicle context. Consequently, since Algorithm 6.1 uses all the possible combinations of fuzzy sets for the input variables to generate the rules of the context-awareness fuzzy system, then it is certain that we have one rule to represent each possible context. Even those contexts where there are no neighboring cars, or where they are far away, are covered by the rules we have obtained.

Finally, as result of the Algorithm 6.2 we have a set of rules, where the antecedent of every rule corresponds to a context representable by our proposed variables; and, the conclusion of every rule indicates the optimal avoidance direction for the particular context.

As mentioned, the validation of our approach is based on a generalization allowing to guide the vehicles towards the optimal direction. Experimental aspects will be considered in the next section to complete our previous demonstrations. We aim to test the generated rules in a variety of collision risks scenarios; however, since manually conceiving test scenarios could be biased and time consuming, we propose the implementation of a random scenario generator where we can test the collision avoidance architecture of Figure 5.7 using the rules generated by the Algorithm 6.2. We describe the details of such scenario generator and the associated experiments in the next section.

6.6 Experiments and results

6.6.1 Scenario generator

We have implemented, on top of the simulator presented in section 5.6, a random scenario generator that uses the environment of Figure 6.1 as a basic pattern. It is conceived to randomly locate a specified number of vehicles in the simulation environment, configure their speeds, and starts the simulation.

For simplification purposes in the scenario configuration, we consider the highway as being the horizontal axis, then the location of vehicles in the simulation is considered, in meters, relative to "*v0*" on this axis. Given that "*v0*" is at the horizontal coordinate 0, then the vehicles behind it have negative coordinates and vehicles ahead of it have positive coordinates; we depict this situation in Figure 6.3.

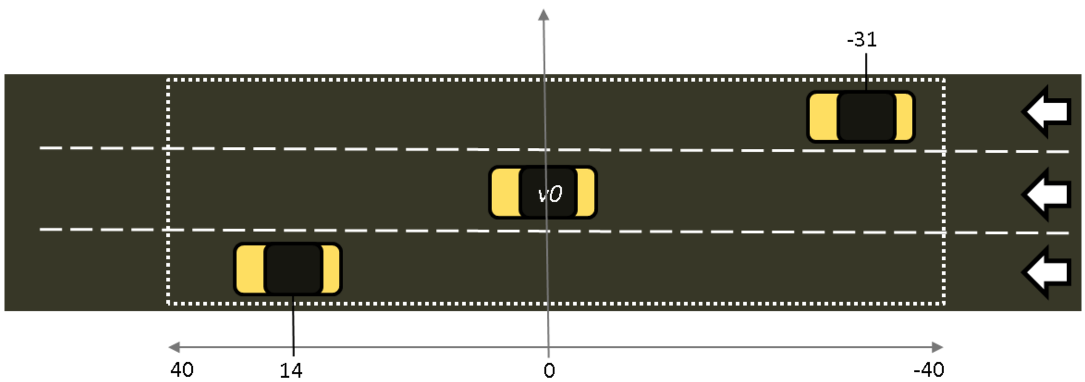

Figure 6.3 Vehicle locations relative to "*v0*"

We want to evaluate the collision avoidance capabilities of "*v0*" when the vehicle in front suddenly brakes because of an unforeseen event on the road. Therefore, as shown in Figure 6.1, the basic configuration of the scenarios to be created by the generator includes "*v0*" and "*v1*" on the same lane.

Before the generator starts, we indicate how many vehicles must appear on each lane, then, it randomly gets, for each vehicle in the scenario, the values the following parameters:

- Location in the scenario.

- Max speed (in kph).

With these parameters configured, the scenario generator starts the simulator and locates the vehicles in their positions. Later, when the simulation starts, all vehicles are centered on their own lanes and have zero speed progressively increasing it up to their own max speeds.

With this tool, we can generate and simulate any number of different random scenarios that will serve as a practical validation of the rules created for the context-awareness fuzzy system. The idea is to visually confirm that the systems within the proposed architecture correctly guide the car away from the danger. In the next section, we describe the experiments and discuss the results.

6.6.2 Validation tests

To validate that the rules generated by the Algorithm 6.2 are generalized for any collision situation, we should evaluate how the collision avoidance system behaves in any scenario. However, testing with all possible configurations would be a long time-consuming task and out of the scope of this research. Thus, in order to cover a broad number of different situations, we have divided the tests into three context cases:

a) No vehicles on the contiguous lanes.
b) One vehicle on one of the contiguous lanes, and
c) Vehicles on both contiguous lanes.

For each case, we have randomly generated 20 different scenarios and ran the corresponding simulations. In all of the cases, the vehicle marked as "*v1*" (from Figure 6.1) is configured to suddenly apply the brakes because of an unforeseen event 6 seconds after the simulation starts, this situation forces "*v0*" to trigger the collision avoidance system and react accordingly.

6.6.2.1 Tests for context case a

This case refers to a scenario where “v0” and “v1” are the only vehicles on the road; therefore, the scenario generator was executed specifying that there should not be vehicles on the lanes contiguous to “*v0*”. In Table 6.8 we present the values of the parameters generated for the 20 simulations made for the first context case.

Table 6.8 Parameters obtained by the scenario generator for the first case tests

Scenario #	Speed of *v0*	Location of *v1*	Speed of *v1*
1	54	20	53
2	36	16	50
3	42	19	40
4	55	19	40
5	44	15	49
6	44	15	50
7	53	35	43
8	52	26	42
9	43	26	39
10	35	19	46
11	54	31	50
12	44	22	45
13	53	22	40
14	43	23	51
15	54	30	50
16	43	16	51
17	40	27	54
18	48	26	41
19	48	35	49
20	36	25	43

To have an idea of how these values are used to build the environment in the simulation, in Figure 6.4 we depict the vehicles placed according to scenario #1 (from Table 6.5) in our simulator.

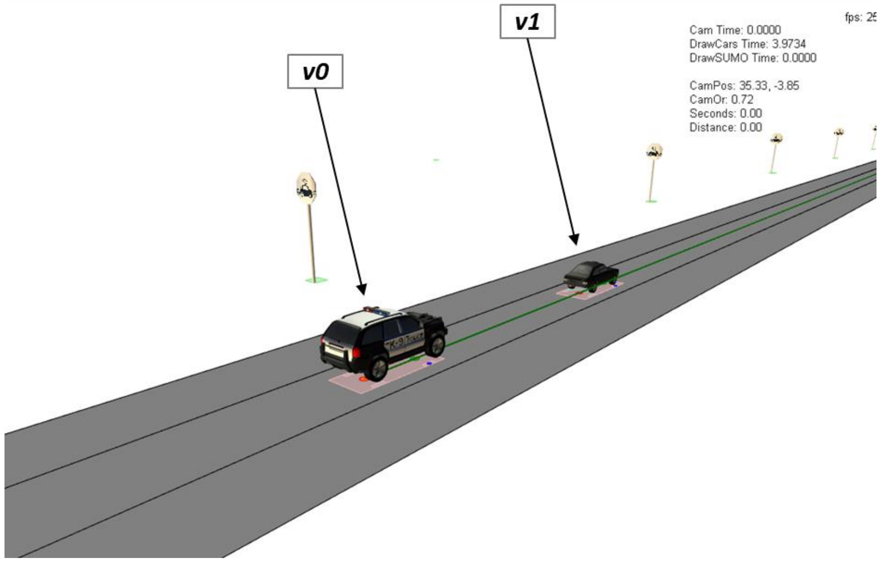

Figure 6.4 Scenario #1 from first case test in simulator

After running the simulations, vehicle "*v0*" successfully avoided the collision in all the scenarios by executing the avoidance maneuver indicated by the context-awareness fuzzy system. In Table 6.9, we summarize the avoidance direction recommendation for the 20 simulations; as we can observe in all the cases the system recommended to avoid by the left lane. In Figure 6.5 we show a snapshot of "*v0*" while it is executing the avoidance maneuver on scenario #1.

Table 6.9 Avoidance direction for the first case tests

# of scenarios	Avoid by the left	Avoid by the right	Avoid by braking
20	20	0	0

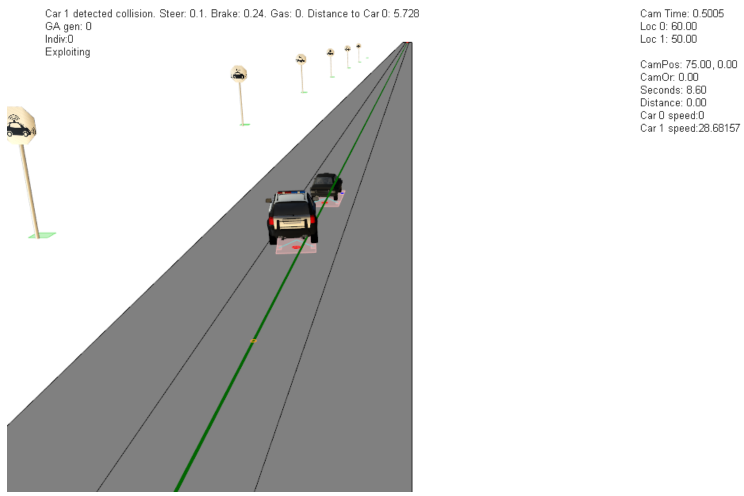

Figure 6.5 *v0* executing avoidance maneuver on Scenario #1

To graphically illustrate how the vehicles perform during the simulation, we present in Figure 6.6 the speeds of both vehicles in scenario #1. It can be observed how they increase their respective speeds (lines orange and blue) up to the max; then, at 6s, *v1* suddenly brakes and its speed rapidly goes to zero. At 7.84s, *v0* starts its avoidance maneuver by changing lanes and reducing its speed until 9.84s. At this moment, it is already on the parallel lane and accelerates again since the collision danger no longer exists. The gray line depicts how the distance between vehicles changes during the simulation.

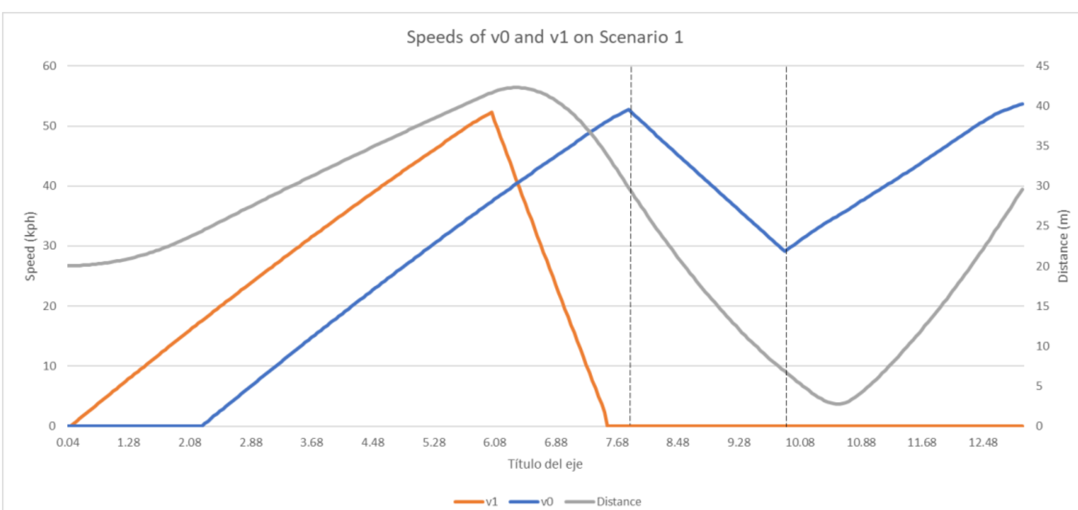

Figure 6.6 Speeds of *v0* and *v1*, and distance between them on scenario #1

6.6.2.2 Tests for context case b

The context case b refers to a collision scenario with a car in one of the two contiguous lanes of "*v0*". So, for this second case, the scenario generator was configured 10 times with a car on the left lane of "*v0*" and 10 times with a car on its right lane. In Table 6.10 we present the values of the parameters generated for these 20 simulations.

Table 6.10 Parameters obtained by the scenario generator for the second case tests

Scenario #	Speed of *v0*	Location of *v1*	Speed of *v1*	Occupied lane	Location of third car	Speed of third car
21	50	13	45	Left	-18	49
22	50	33	50	Left	-38	40
23	38	33	43	Left	-26	45
24	44	21	36	Left	-5	43
25	46	32	43	Left	-12	48
26	38	15	54	Left	-32	46
27	53	32	49	Left	-29	47
28	48	30	54	Left	-9	38
29	52	29	42	Left	-15	53
30	54	21	53	Left	-12	51
31	53	16	49	Right	-19	38
32	54	15	48	Right	-10	47
33	42	30	54	Right	-29	51
34	51	20	37	Right	-36	47
35	37	16	52	Right	-19	39
36	40	25	40	Right	-37	41
37	42	26	52	Right	-19	40
38	47	35	55	Right	-37	51
39	40	34	36	Right	-22	38
40	41	23	46	Right	-36	38

In Figure 6.7 we depict the vehicles placed in the simulation environment according to the parameters of scenarios #21 and #31 (from Table 6.10).

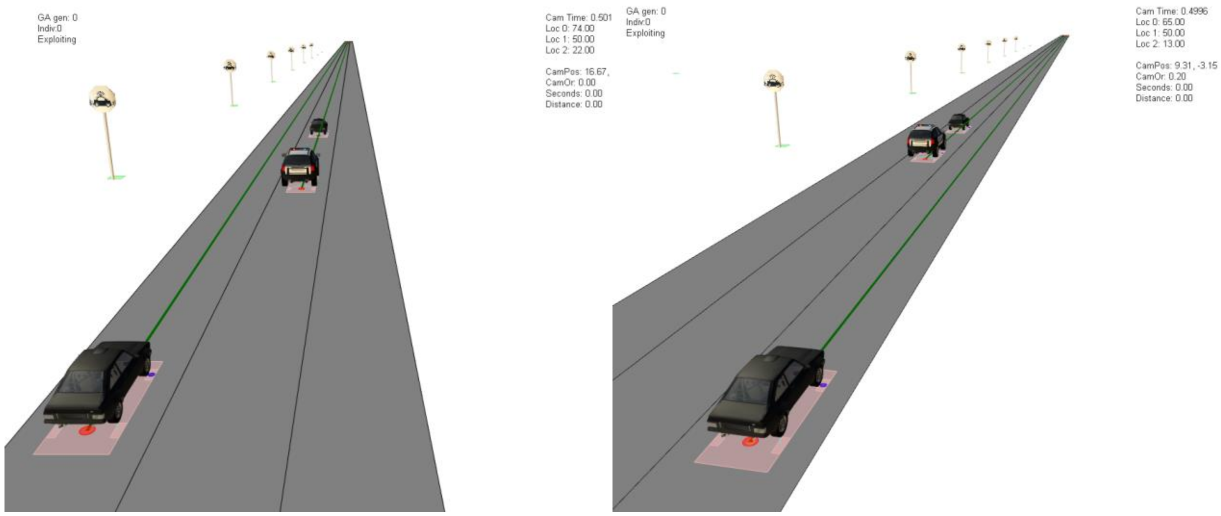

Figure 6.7 Scenario#21 (left) and #31 (right) from second case test in simulator

After running the simulations for the configurations of Table 6.10, the vehicle "*v0*" successfully avoided the collision in all the scenarios by executing the avoidance maneuver indicated by the context-awareness fuzzy system. In Table 6.11, we summarize the avoidance direction recommendation for the 20 simulations; as we can observe, in three of the scenarios where a car was on the left lane the avoidance direction was to the left, and in the other 7 the avoidance was executed to the right. It is worth noting that an avoidance maneuver to the left, while a car is on that lane, is valid if the distance between cars allow an execution without collision risk.

Table 6.11 Avoidance direction for the second case tests

Scenario #	Avoid by the left	Avoid by the right	Avoid by braking
21-30	3	7	0
31-40	10	0	0

The left side of Figure 6.8 depicts scenario #21, where *v0* changes lanes to the right while *v2* is on the left lane. The right side of Figure 6.8 depicts scenario #31 where *v0* changes lanes to the left while *v2* is on the right lane.

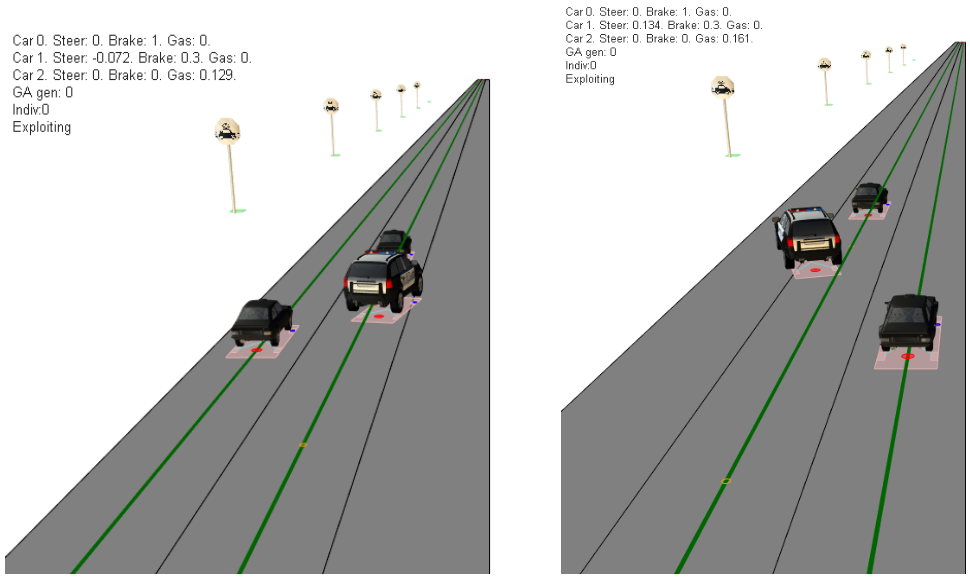

Figure 6.8 *v0* executing avoidance maneuver on Scenarios #21 (left) and #31 (right)

To graphically illustrate how the vehicles perform during the simulation, we present in Figure 6.9 the speeds of all vehicles in scenario #21, as well as the distance between *v0* and *v1* (in gray). As in the previous cases, it can be observed that all vehicles progressively increase their speeds, and at 6s *v1* suddenly brakes (orange line); then, at 8.12s, *v0* starts its avoidance maneuver by changing lane to the right and reducing its speed (blue line) until 10.72s when it is on the parallel lane and accelerates again since the collision danger no longer exists. *v2* continues on its lane at its max speed without interference. Figure 6.10 depicts the same information for scenario #31 in which *v0* changes lanes to the left, and *v2* continues on the right lane. It is clear that in both cases cars do not collide and they continue their paths as planned.

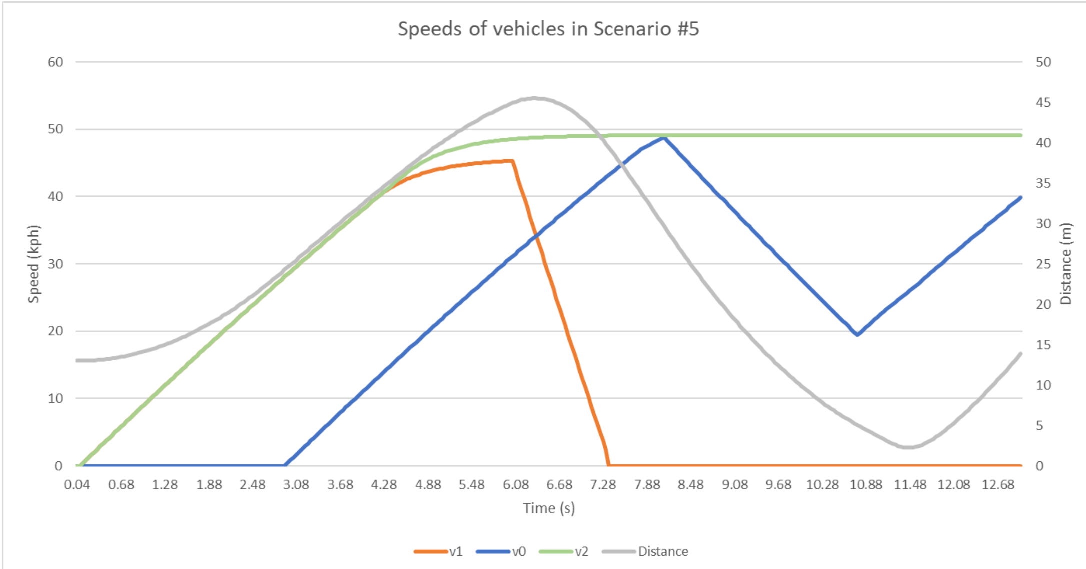

Figure 6.9 Speeds of *v0, v1* and *v2*, and distance between *v0-v1* on scenario #21

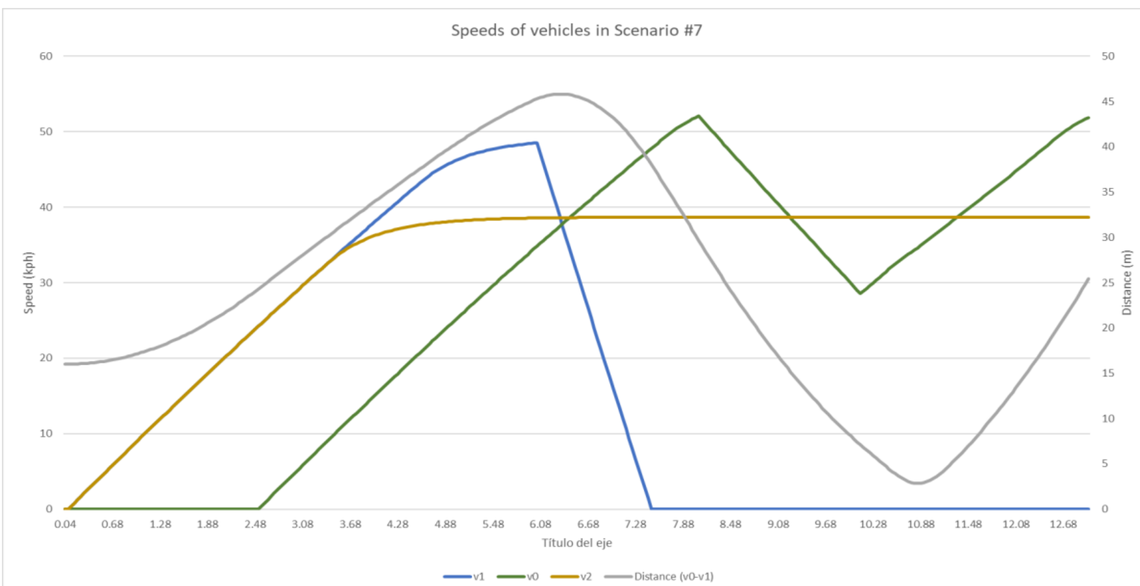

Figure 6.10 Speeds of *v0, v1* and *v2*, and distance between *v0-v1* on scenario #31

6.6.2.3 Tests for context case c

The third context case refers to a collision scenario with a car in both of the contiguous lanes of "*v0*". So, for this case, we ran again 20 simulations with the scenario generator, specifying that there should be vehicles on the left and right lanes. The values of the parameters generated for these runs are presented in Table 6.12.

Table 6.12 Parameters obtained by the scenario generator for the third case tests

Scenario #	Speed of *v0*	Location of *v1*	Speed of *v1*	Location of third car (left)	Speed of third car	Location of fourth car (right)	Speed of fourth car
41	54	10	52	-20	58	-16	47
42	45	16	43	-33	42	-34	46
43	46	17	55	-21	35	-29	53
44	38	21	48	-17	40	-15	49
45	43	18	55	-28	51	-38	48
46	38	24	43	-34	52	-14	54
47	45	16	43	-33	42	-44	36
48	51	22	40	-14	35	-13	48
49	44	15	40	-19	44	-14	51
50	54	18	41	-22	43	-12	55
51	52	20	50	-15	43	-19	55
52	45	30	55	-16	45	-20	50
53	47	35	42	-33	37	-39	50
54	44	27	54	-8	55	-6	42
55	50	30	44	-8	37	-18	35
56	52	15	41	-26	41	-21	54
57	54	35	49	-25	55	-6	55
58	41	28	53	-31	35	-20	41
59	51	18	54	-8	55	-28	41

Figure 6.11 depicts the vehicles placed in the simulator according to the configuration parameters of scenario #41 (from Table 6.12).

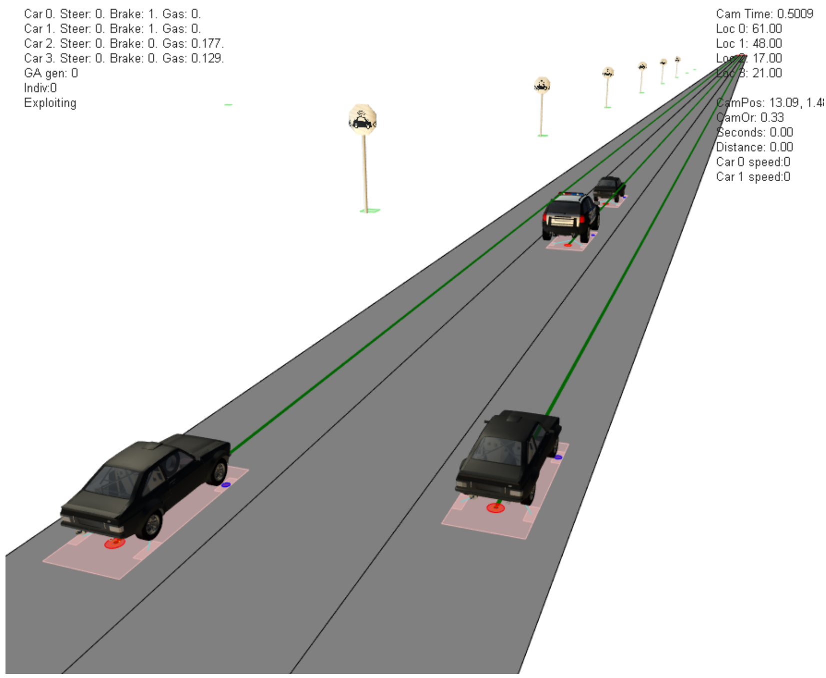

Figure 6.11 Scenario #41 from third case test in simulator

After running the simulations, vehicle "*v0*" successfully avoided the collision in all the scenarios by executing the avoidance maneuver indicated by the context-awareness fuzzy system. In Table 6.13, we summarize the avoidance direction indicated by the system for each of the 20 scenarios of Table 6.12.

Table 6.13 Avoidance direction for the third case tests

# of scenarios	Avoid by the left	Avoid by the right	Avoid by braking
20	7	4	9

Figure 6.12 depicts *v0* applying the brakes to avoid the collision with *v1* in scenario #41, since vehicles in both contiguous lanes are too close this is the only option to take. In Figure 6.13, we present the speeds of all vehicles in the simulation for scenario #41, as well as the distance between *v0* and *v1* (in gray). As the simulation starts, vehicles have zero speed and progressively increase

it, and at 6s *v1* suddenly brakes; then, at 7.88s, *v0* starts its avoidance maneuver by applying the brakes until zero speed. It can be seen in Figure 6.13 that *v2* and *v3* continue with their respective speeds. Also, since *v0* and *v1* are on the same lane, a safe distance of at least 6m is kept between them at the end of the avoidance maneuver.

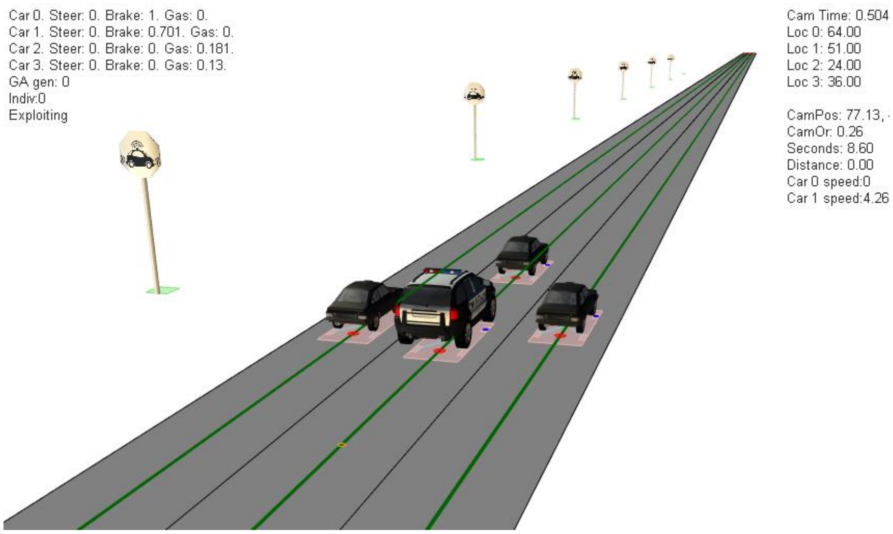

Figure 6.12 *v0* executing avoidance maneuver (stop) on Scenario #41

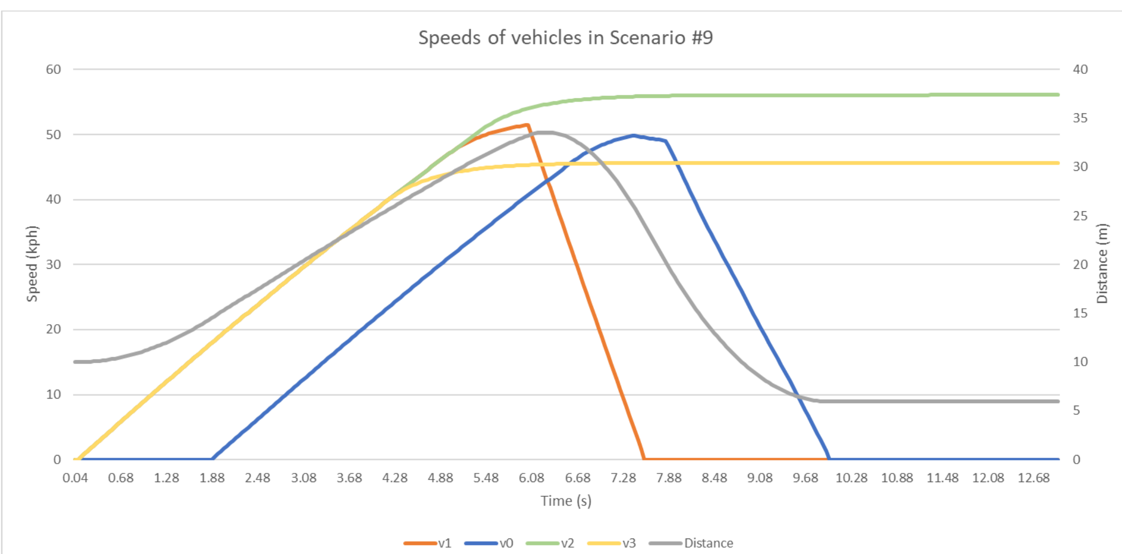

Figure 6.13 Speeds of *v0*, *v1*, *v2* and *v3*, and distance between *v0-v1* on scenario #41

It is worth noticing here that a relevant result of the dissemination of the vehicle's intention is the collaboration when a collision situation arises. For instance, we take scenario #47 from Table 6.12 and depict its initial setup on the simulator in Figure 6.14. Even if there is a vehicle on the right lane, *v0* decides to go that way because there is enough space to safely execute the maneuver. By executing this action, it informs *v3* (vehicle on the right lane) of its intention to invade the right lane, then *v3* triggers its own collision avoidance system to collaborate with the avoidance maneuver of *v0* and prevent an impact with it. We depict this situation in Figure 6.15, where *v0* is going to the right lane, and *v3* is braking to let *v0* pass.

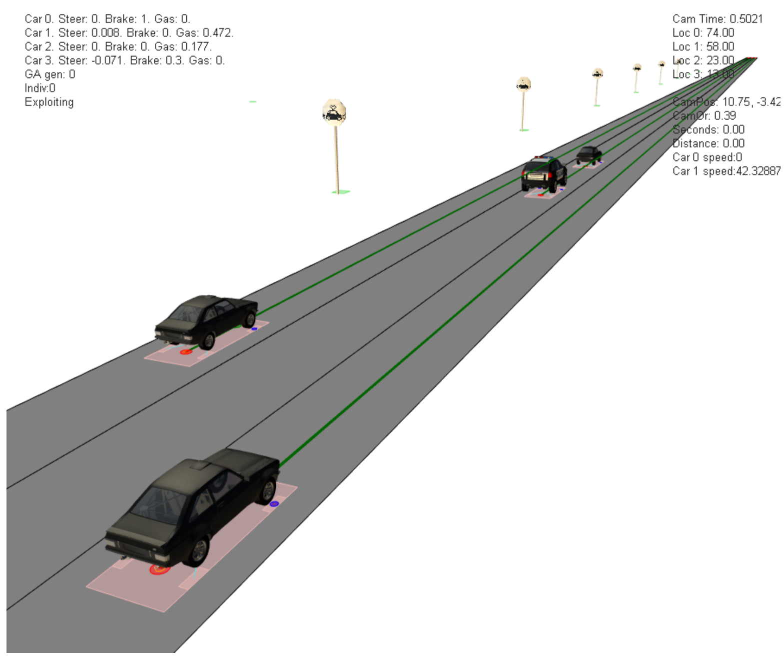

Figure 6.14 Scenario #47 from third case test in simulator

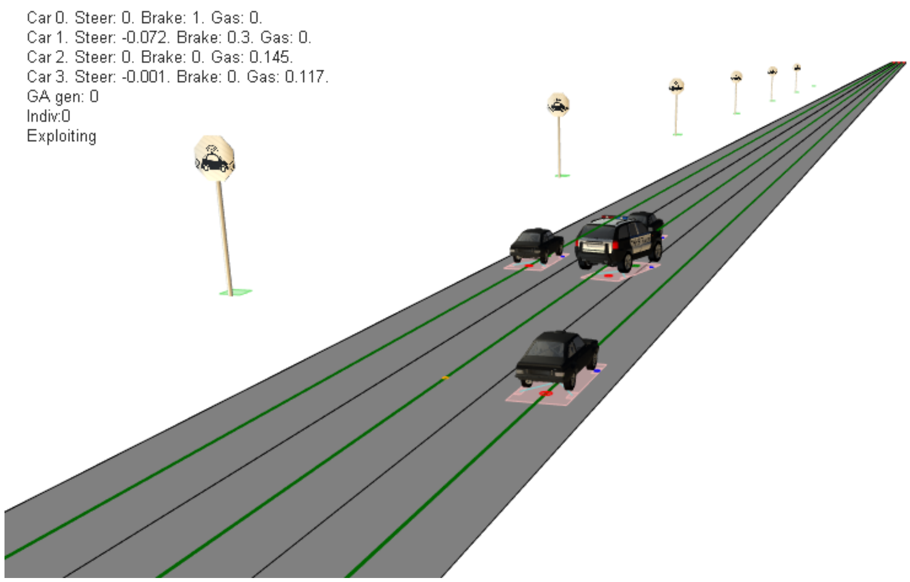

Figure 6.15 *v0* and *v1* collaborating to avoid collision on scenario #47

To graphically illustrate the situation of scenario #47, we show the speeds of vehicles during this simulation in Figure 6.16. It can be observed that at 6s *v1* suddenly brakes, then *v0* at 7.84s starts braking and changes lanes to avoid colliding with *v1*. Later, as *v0* is sending its intention information to other vehicles in the scenario, *v3* collaborates with the avoidance maneuver of *v0* by applying the brakes at 10.64s; it brakes and waits for *v0* to regain speed and distance. In Figure 6.17 we present the evolution of the distance between *v0* and *v1*, as well as the distance between *v0* and *v3*, during the simulation of scenario #47.

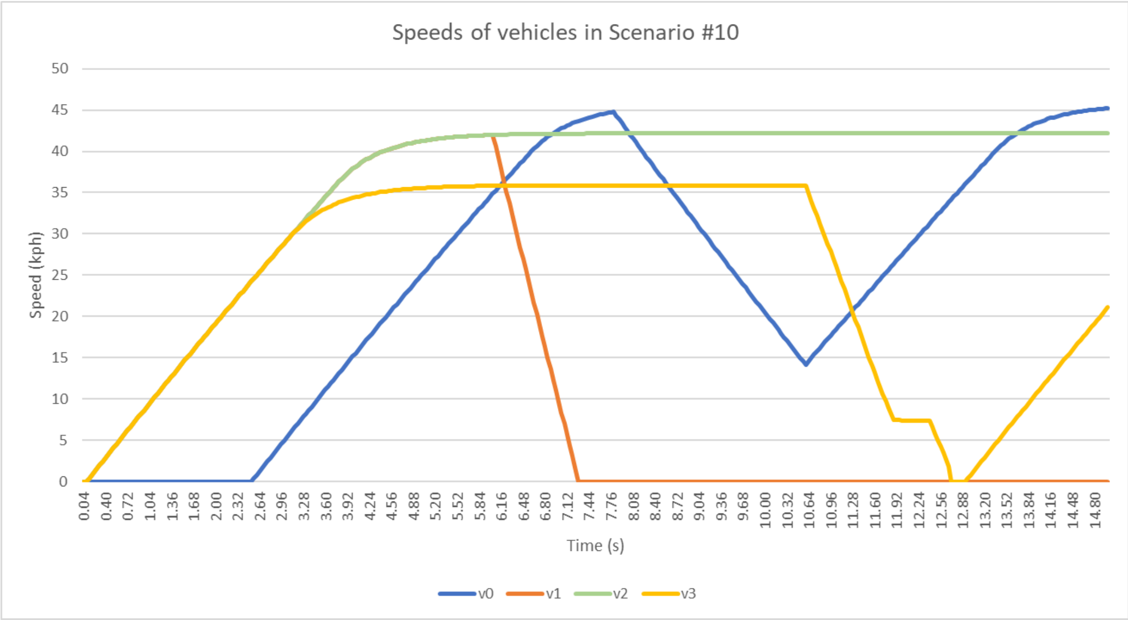

Figure 6.16 Speeds of *v0*, *v1*, *v2* and *v3* on scenario #47

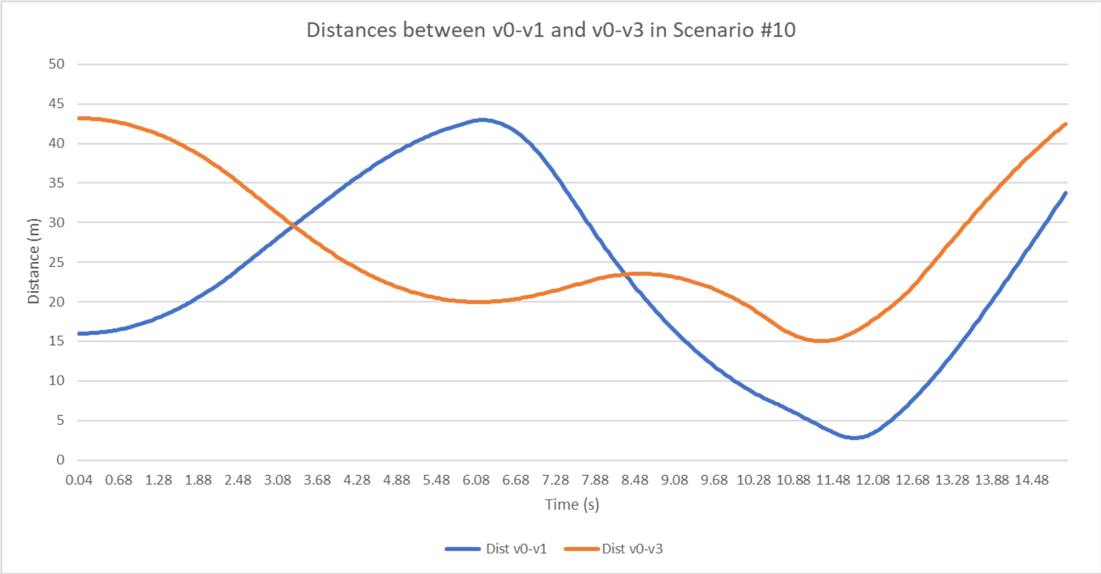

Figure 6.17 Distances between *v0-v1* and *v0-v3* on scenario #47

Chapter VII

Conclusions and Future Work

7.1 Final remarks

Recent advances in communication technologies built inside vehicles offer an additional source of information, valuable since without it vehicles know only what their ego-sensors can capture; such technologies allow them to be aware of what is happening beyond the range of their sensors. Thus, connected vehicles provide further driving enhancement possibilities for the driver and the occupants, thanks to the use of traffic, weather and entertainment information; furthermore, being able to share safety-related information is an advantage when dealing with time-critical dangerous situations, especially for autonomous and semi-autonomous vehicles.

The integration of communication capabilities with autonomous driving technologies opens the space for research in solutions for cooperative collision avoidance. This is a research area with increasing interest; currently, there is a world attention on this domain, denoting its importance and the need of innovation in it. For instance, the United Nations Assembly adopted a resolution of a global plan of action for road safety (2011-2020); such an initiative further validates the concentrations of efforts on the matter.

The availability of information from multiple sources is useful for an AV, since it can integrate it to what it already knows and produce cooperative solutions to problematic scenarios that involve multiple vehicles on the road. However, making several vehicles agree on how to solve a particular risky situation is still a problem to be addressed; the variety of hardware sensors and danger detection algorithms complicates this necessary agreement.

While driving autonomously, AVs must decide if a given situation represents a menace for the occupants or not, and decide upon; sometimes they will face situations which response time is critical, like a vehicle braking unexpectedly, or another one approaching with collision trajectory. In such scenarios, the AV must generate an action plan and execute it immediately in order to avoid dangerous consequences. Moreover, since the AV shares the roads with other vehicles, a cooperative approach must consider their actions as part of a more complex collective solution.

Although metaheuristics have been considered in the research on ITS, they are focused on deliberative solutions for long-term goals. More recently, there is also a tendency to study deep-learning techniques in this field for computer vision and automatic classification. Our fuzzy proposal is complementary to those solutions, it is a short-term decision-making system that could eventually integrate and take advantage of their outputs. AVs will require of multiple parallel and redundant processes to fully understand the situation on the road and produce optimal results.

In this research work, we consider that vehicular networks resemble multi-agent systems where vehicles are represented by intelligent agents. Thanks to on-board sensors, the agents controlling autonomous vehicles recognize the presence of other vehicles in the street; moreover, they can share location and speed data as well. The cognitive multilayer architecture proposed in this research integrates environment-related knowledge with intention of neighbouring cars in order to anticipate the possibility of a collision and to produce cooperative maneuvers to avoid it. Faced with unsafe circumstances, the agent can recognize what is happening and generate an action plan; one that solves its individual situation and the collective one. Taking advantage of context recognition and communication capabilities to share knowledge, our proposal can achieve such cooperative goal.

7.2 Contributions

At the beginning of the present research we identified and explained part of the current problematic and recent challenges in the Intelligent Transportation Systems domain, more specifically those related to collision avoidance in autonomous vehicles. The following is a list of the main limitations found in the state of the art:

- A unique and common data structure to store and share information about the roads and the vehicles.
- A simulation environment to safely test and observe the execution of avoidance maneuvers in collision scenarios.
- Consider the contextual conditions to make informed decisions in collision risk situations.
- A decentralized way to collaboratively resolve collision risk situations on the road.

These challenges were the main focus of this research, and as result we have proposed solutions to minimize or eliminate their effects; such solutions have also been selected for publication in refereed journals and international conferences. In the following paragraphs, we discuss how our proposal addresses each problematic.

7.2.1 A unique and common data structure to store and share information about the roads and the vehicles

A common ontology to express knowledge related to a vehicular environment was proposed; this will allow that multiple vehicles have a unique understanding of the situation of the road they are sharing; such as the risk of a collision, given the current state of their environment. The proposal serves as a compatible way to communicate information among the vehicles in a VANET; this contributes to the interoperability in a diversity of vehicles because the dissemination of messages, using a common ontology, will guarantee that all vehicles recognize and agree on a single interpretation of what is happening in the world around them. A preliminary version of this ontology was presented in the 15th IFAC Symposium on Information Control in Manufacturing [128].

7.2.2 A simulation environment to safely test and observe the execution of avoidance maneuvers in collision scenarios

As part of the development of this research we have created a collision scenario simulation environment. This new simulator was designed as a tool to support safe experimentation of collision avoidance approaches; it allows us to simulate fixed or random scenarios with AVs. More importantly, the opportunity to visually evaluate the execution and outcome of collision avoidance maneuvers gives the researcher a realistic point of view of the performance of prospect approaches. The design and implementation of this simulator, as a supporting tool for the research of cooperative driving solutions for the collision avoidance problem, was published in the IFAC-Papers On Line journal as the proceedings of the 15th IFAC Symposium on Information Control in Manufacturing [126].

7.2.3 Consider the contextual conditions to make informed decisions in collision risk situations

We have designed and implemented an intelligent agent architecture that uses a series of stacked fuzzy systems to assess the collision risk using contextual information from the close environment, produce an avoidance strategy, and generate the controls for the vehicle actuators. A set of rules, for the context-awareness fuzzy system, was generated with a generalization approach in order to consider all the context situations that can be represented by our proposed variables. We showed that with this architecture, an AV is enabled to consider the state of other vehicles on the road when deciding how to avoid a collision situation on a highway.

7.2.4 A decentralized way to collaboratively resolve collision risk situations on the road

Another contribution of this research is the conception of a cooperative decentralized approach to solve a collision situation based on context-awareness. Even if there exist approaches for cooperative collision avoidance, our proposal is innovative because it integrates the intention of the vehicle. At the core of the proposed architecture, when a risk of collision is detected, the vehicle informs its neighbors how it intends to avoid it; we have shown how notifying the vehicle's intention reduces the reaction time of other vehicles. By taking advantage of communication capabilities onboard vehicles, we can support a collaborative environment where vehicles non-directly involved in the collision can actively perform avoidance maneuvers that assist from the collective perspective. Results of this cooperative approach were presented at the ITS World Congress 2017 [129].

7.3 Future Work

Based on the results obtained in this research, we have identified four main directions for future development of our proposal: ITS applications, knowledge base integration, communication performance, and information security.

7.3.1 ITS applications

Among the applications that we are interested in further study is a cooperative approach for collision avoidance in emergency vehicles. When responding to an emergency, this type of vehicles has the possibility to legally break some traffic rules, such as exceeding the maximum speed, crossing red traffic lights, driving on the street shoulder and even on contrary way lanes. Obviously, in an emergency situation, this irregular behavior is accepted because it could potentially save lives; however, it introduces a new level of danger for the vehicles in the vicinity of the emergency responder. Advances in this direction have already been started, some preliminary results of simulating the use of connected infrastructure with digital panels, and variated percentage of CV presence, were presented as a seminar during the event *Semaine de la recherche* (*Research week*) at *Université du Québec en Outaouais* (UQO), with the attendance of ITS and road safety professionals, including representatives of the Gatineau Police Department [130].

7.3.2 Knowledge base integration

Currently, entities in the proposed ontology keep information about the vehicle's current state and its close environment, including other vehicles and the road infrastructure. However, it lacks of knowledge related to the controls of its actuators; therefore, it is not possible to query the ontology for the necessary actuators' avoidance actions in a collision risk situation. Entities to model the basic actions of the vehicle actuators could be added to current ontology design. Instances of such entities will keep information about the valid values for the components of an evasive action, namely the acceleration/brake and steering wheel angle values. Later, it will be possible to incorporate an ontology-based avoidance process to obtain optimal values that avoid a collision in a given scenario.

7.3.3 Communication performance

In this research, we made abstraction of the communication layer, and assumed it is non-faulty. However, this is not always true in the real world; there are a number of factors that can affect the quality of the communication, such as: weather conditions, line of sight, reflective surfaces in buildings and big vehicles, among others. Since a cooperative collision avoidance system relies on the information sent by neighboring vehicles, malfunctioning communication elements or missing data are of major concern.

It is clear that further research is required to estimate the possibility of failures in the communication, and simulate its effects on the performance of the collision avoidance system. Moreover, the protocols used to route data packages in a VANET play a relevant role in the performance of the network, and therefore in the effectiveness of the collision avoidance system. In highly occupied environments, the data volume can be significantly increased, producing bottlenecks and delays in data delivery, which can be translated into vehicles dealing with obsolete information. Thus, it is necessary to study how the system can be adapted to tolerate such faulty situations without loosing reliability nor augmenting the risk of danger. In this sense, we have already started communication with Prof. M. Shawky from *Université de Technologie de Compiègne* in France, who has a research team working on this domain, to collaborate in a shared project.

7.3.4 Information security

Finally, information security and trust is another domain we aim to explore in future developments. As part of this research, we assumed that information coming from other vehicles is valid and is true; in other words, we trust that the source is sending real location and context information. Nevertheless, as in other networks, in a VANET there could be malicious actors trying to inject false data in order to take advantage or priority on the road. Hence, the detection and discard of such false messages is also a problematic to be addressed. We believe that distributed consensus protocols, such as the Byzantine algorithms, to detect adversaries in the network, are an interesting line of action for this problematic.

7.4 List of publications

- I. Benyahia, G. **Colmenares**, M. Shawky, M. Zaremba, "Hierarchical Architecture for Connected Vehicles Collision Avoidance", in ITS Canada Annual Conference and General Meeting (ACGM 2014), pp. 123-129, 2014.
- G. **Colmenares**, F. Halal, and M. Zaremba. "Ant colony optimization for data acquisition mission planning.", Management and Production Engineering Review, vol. 5, no. 2, pp. 3-11, 2014.
- G. **Colmenares**, and I. Benyahia, "Cooperative Driving Experiments Based on a 3D Components Environment.", IFAC-PapersOnLine, vol. 48, no. 3, pp. 2351-2355, 2015.
- G. **Colmenares**, I. Benyahia, M. Shawky, and M. Zaremba. “Intelligent Agents for Cooperative Driving: Challenges and Solutions,” presented at the 15th IFAC Symposium

on Information Control Problems in Manufacturing (INCOM 2015), Ottawa, Canada, pp.76-84, 2015.

- I. Benyahia, G. **Colmenares**, and F. Bellavance, "From Highways Risk Monitoring Process to Intelligent System for Highways Safety Calibration for Autonomous Vehicles", AQTR - Routes et Transports, vol. 46, no. 2, pp. 52-57, 2017.
- G. **Colmenares**, and I. Benyahia. "Role of Vehicle's Intention on the Optimization of Collision Avoidance Strategies for Cooperative Driving.", in 24th Annual ITS World Congress 2017, Montreal, Canada, 2017.

References

[1] R. J. Weiland, and L. B. Purser, "Intelligent Transportation Systems", Transportation in the new millennium, 2000.

[2] G. Dimitrakopoulos, and P. Demestichas, "Intelligent Transportation Systems", IEEE Vehicular Technology Magazine, vol. 5, no 1, 77-84, 2010.

[3] G. Dimitrakopoulos, "Intelligent Transportation Systems based on Internet-Connected Vehicles: Fundamental Research Areas and Challenges", 11th IEEE International Conference on ITS Telecommunications (ITST), pp. 145-151, 2011.

[4] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications." IEEE Transactions on Intelligent Transportation Systems, vol. 3, no 3, pp. 155-161, 2002.

[5] C-L. Lee, C-Y. Huang, T-Ch. Hsiao, C-Y. Wu, Y-C. Chen, I.-C. Wang, "Impact of Vehicular Networks on Emergency Medical Services in Urban Areas", International journal of environmental research and public health, vol 11, no 11, 11348-11370, 2014.

[6] K. Basak, S. N. Hetu, Z. Li,C. Lima, H. Loganathan, T. Toledo, et al. "Modeling reaction time within a traffic simulation model", In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 302-309, 2013.

[7] J. Sewall, D. Wilkie, M. C. Lin, "Interactive hybrid simulation of large-scale traffic". In ACM Transactions on Graphics (TOG), vol. 30, no 6, p. 135, 2011.

[8] R. Stanica, E. Chaput, A. L. Beylot, "Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations". Computer Networks, vol. 55, no 14, 3179-3188, 2011.

[9] D. Muoio, “Here are all the companies racing to put driverless cars on the road by 2020”. [online] Tech Insider. Available at: http://www.techinsider.io/google-apple-tesla-race-to-develop-self-driving-cars-by-2020-2016-4 [Accessed May 2016].

[10] G. Dimitrakopoulos, “Intelligent Transportation Systems based on Internet-Connected Vehicles: Fundamental Research Areas and Challenges”, 11th International Conference on ITS Telecommunications, pp. 145-151, 2011.

[11] R. J. Weiland and L. Baughman P., “Intelligent Transportation Systems”, Transportation in the new millennium, 2000.

[12] G. Dimitrakopoulos and P. Demestichas, “Intelligent Transportation Systems”, IEEE Vehicular Technology Magazine, vol. 5, no 1, pp. 77-84, 2010.

[13] U.S. Department of Transportation, “Traffic Safety Facts, Research Note”, DOT HS 812 260, 2016.

[14] Government of Alberta, “Government strengthens penalties for distracted driving”, [online] Alberta government, Available at: http://www.alberta.ca/release.cfm?xID=3899083E9C16E-D24D-5957-C49B1B19A92F2455, 2015 [Accessed June 2016].

[15] C. Wang, M. Quddus, and S. Ison. "Impact of traffic congestion on road accidents: a spatial analysis of the M25 motorway in England." Accident Analysis & Prevention, vol. 41, no 4, pp. 798-808, 2009.

[16] G-L. Chang, and H. Xiang. "The Relationship between Congestion Levels and Accidents". No. MD-03-SP 208B46, 2003.

[17] Eurostat, Road safety statistics at regional level, [online] Eurostat, Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/Road_safety_statistics_at_regional_level. [Accessed June 2016].

[18] R. Retting, B. Persaud, P. Garder, and D. Lord, “Crash and injury reduction following installation of roundabouts in the United States”, American journal of public health, vol. 91, no 4, p. 628, 2001.

[19] U.S. Department of Transportation, National highway Traffic Safety Administration, “National motor Vehicle Crash Causation Survey”, Report to congress, DOT HS 811 059, 2008.

[20] P. Falcone, F. Borrelli, J. Asgari, H.E.Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems”, IEEE Transactions on control systems technology, vol. 15, no 3, pp. 566-580, 2007.

[21] J. Lee and B. Park, “Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm under the Connected Vehicles Environment”, IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1, pp. 81-90, march 2012.

[22] C. Chan, “Connected Vehicles in a Connected World”, IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT), pp. 1-4, 2011.

[23] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar and B. Litkouhi, “Towards a Viable Autonomous Driving Research Platform”, IEEE Intelligent Vehicles Symposium (IV), pp. 763-770, 2013.

[24] M. Abdul-Hak, N. Al-Holou, U. Mohammad, M. A. Tamer and M. Arafat, “ITS Based Methodology to Reduce Energy Consumption and Emissions”, IEEE Transportation Electrification Conference and Expo (ITEC), pp. 1-7, 2012.

[25] B. Park, K. Malakorn and J. Lee, “Quantifying Benefits of Cooperative Adaptive Cruise Control Toward Sustainable Transportation System”, Center for Transportation Studies, University of Virginia, Charlottesville, VA, 2011.

[26] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, “Connected vehicles: solutions and challenges.” IEEE Internet of Things Journal, vol. 1, no 4, pp. 289-299, 2014.

[27] C. Desjardins and B. Chaib-draa, “Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach,” IEEE Transactions on Intelligent Transportation Systems, vol. 12, no 4, pp. 1248-1260, 2011.

[28] G. M. Arnaout, M. T. Khasawneh, J. Zhang and S. R. Bowling, “An IntelliDrive Application for Reducing Traffic Congestions using Agent-Based approach”, Proceedings of the IEEE Systems and Information Engineering Design Symposium (SIEDS), pp. 221-224, 2010.

[29] “The connected vehicle”, World Road Association (PIARC), 2012.

[30] Y. Toor, P. Mühlethaler and A. Laouiti, “Vehicle Ad Hoc Networks: Applications and Related Technical Issues”, IEEE Communications Surveys & Tutorials, vol. 10, no 3, pp. 74-88, 2008.

[31] F. Li and Y. Wang, “Routing in Vehicular Ad Hoc Networks: A Survey”, IEEE Vehicular Technology Magazine, vol. 2, no 2, pp. 12-22, 2007.

[32] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, “Traffic Information System: A Lightweight Geocast-based Piggybacking Strategy for Cooperative Awareness in VANET,” presented at the IEEE International Conference on Consumer Electronics (ICCE), pp. 614-615, 2013.

[33] M. L. Sichitiu, and M. Kihl, “Inter-vehicle communication systems: a survey”, IEEE Communications Surveys & Tutorials, vol. 10, no 2, pp. 88-105, 2008.

[34] M. Nekovee, and B. B. Bogason, “Reliable and efficient information dissemination in intermittently connected vehicular adhoc networks”, IEEE 65th Vehicular Technology Conference, (VTC), pp. 2486-2490, 2007.

[35] M. Rondinone, J. Gozalvez, J. Leguay, and V. Conan, “Exploiting Context Information for V2X Dissemination in Vehicular Networks”, IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1-9, 2013.

[36] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, “Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions”, IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 584-616, 2011.

[37] S. Zeadally, R. Hunt, Y. S. Chen, A. Irwin, and A. Hassan, “Vehicular ad hoc networks (VANETS): status, results, and challenges”, Telecommunication Systems, vol. 50, no 4, pp. 217-241, 2012.

[38] G. Dimitrakopoulos, “Intelligent transportation systems based on internet-connected vehicles: Fundamental research areas and challenges”, IEEE 11th International Conference on ITS Telecommunications (ITST), pp. 145-151, 2011.

[39] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, “A survey of inter-vehicle communication protocols and their applications”, IEEE Communications Surveys & Tutorials, vol. 11, no 2, pp. 3-20, 2009.

[40] D. N. Cottingham, I. J. Wassell, and R. K. Harle, “Performance of IEEE 802.11a in vehicular contexts”, IEEE 65th Vehicular Technology Conference (VTC), pp. 854-858, 2007.

[41] S. Yousefi, M. S. Mousavi and M. Fathy, “Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives”, IEEE 6th International Conference on ITS Telecommunications Proceedings, pp. 761-766, 2006.

[42] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, “A Survey of Inter-Vehicle Communication Protocols and Their Applications,” IEEE Communications Surveys & Tutorials, vol. 11, no 2, pp. 3-20, 2009.

[43] R. Kohlhaas, T. Schamm, D. Lenk, and J. M. Zöllner, ”Towards driving autonomously: Autonomous Cruise Control in Urban Environments”, IEEE Intelligent Vehicles Symposium Workshops, pp. 109-114, 2013.

[44] J. Hu, L. Kong, W. Shu, and M. Wu, “Scheduling of Connected Autonomous Vehicles on Highway Lanes”, Global Communications Conference (GLOBECOM), Wireless Networking Symposium, pp. 5556-5561, 2012.

[45] J. E. Naranjo, C. González, R. García, and T. de Pedro, “Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver”, IEEE Transactions On Intelligent Transportation Systems, vol. 9, no. 3, pp. 438-450, 2008.

[46] Q. Baig, “Fusion de Données Multi Capteurs pour la Détection et le Suivi d’Objets Mobiles à Partir d’un Véhicule Autonome”. Thèse pour Obtenir le Grade de Docteur de l’Université de Grenoble, 2012.

[47] S. A. Shaikh and P. Krishnan, “A Framework for Analysing Driver Interactions with Semi-Autonomous Vehicles”, Electronic Proceedings in Theoretical Computer Science (EPTCS), vol. 105, pp. 85–99, 2012.

[48] M. Parent, “Automated vehicles: Autonomous or connected?”, IEEE 14th International Conference on Mobile Data Management, vol. 1, pp. 2-2, 2013.

[49] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta, “A cooperative approach to traffic congestion detection with complex event processing and VANET,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no 2, pp. 914-929, 2012.

[50] R. Sengupta, S. Rezaei, S. E. Shladover, D. Cody, S. Dickey, and H. Krishnan, “Cooperative Collision Warning Systems: Concept Definition and Experimental Implementation,” Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, vol. 11, no 3, pp.143-155, 2007.

[51] S. Behere, M. Törngren, and D.-J. Chen, “A reference architecture for cooperative driving,” Journal of Systems Architecture, vol. 59, no 10, pp. 1095-1112, 2013.

[52] C. Guo, N. Wan, and S. M. M. Yang, “Self-defensive Coordinated Maneuvering of an Intelligent Vehicle Platoon in Mixed Traffic,” presented at the 15th International IEEE Conference on Intelligent Transportation Systems, pp.1726-1733, 2012.

[53] M. R. I. Nieuwenhuijze, T. Keulen, S. Öncü, B. Bonsen, and H. Nijmeijer, "Cooperative Driving with a Heavy-Duty Truck in Mixed Traffic: Experimental Results," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no 3, pp.1026-1032, 2012.

[54] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A Survey of Inter-Vehicle Communication Protocols and Their Applications", IEEE Communications Surveys & Tutorials, vol. 11, no 2, pp. 3-20, 2009.

[55] L. Güvenç, I.M. Uygan, K. Kahraman, R. Karaahmetoglu, I. Altay, M. Sentürk,, M.T. Emirler, A.E. Karci, B.A. Güvenç, E. Altug, M.C. Turan. "Cooperative adaptive cruise control implementation of team mekar at the grand cooperative driving challenge". IEEE Transactions on Intelligent Transportation Systems, vol. 13, no 3, pp. 1062-74, 2012.

[56] J. Wu, A. Abbas-Turki, and A. E. Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," in Applied Intelligence, Springer, Ed. vol. 37, pp. 207-222, 2012.

[57] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no 1, pp. 81-90, 2012.

[58] R. Kohlhaas, T. S. D. Lenk, and J. M. Zöllner, "Towards driving autonomously: Autonomous Cruise Control in Urban Environments," presented at the IEEE Intelligent Vehicles Symposium, pp.109-114, 2013.

[59] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowsk, H. Kawazoe, & M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," IEEE Transactions on Intelligent Transportation Systems, vol 15, no 1, pp. 296-305, 2014.

[60] P. Schito and F. Braghin, "Numerical and Experimental Investigation on Vehicles in Platoon," SAE International Journal of Commercial Vehicles, vol. 5, pp. 63-71, 2012.

[61] M. Maile, Q. Chen, G. Brown, and L. Delgrossi, "Intersection Collision Avoidance: From Driver Alerts to Vehicle Control", In IEEE 81st Vehicular Technology Conference (VTC Spring), pp. 1-5, 2015.

[62] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative Collision Avoidance at Intersections: Algorithms and Experiments", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no 3, pp. 1162-1175, 2013.

[63] G. Rodrigues, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing", IEEE International Conference on Intelligent Transportation Systems (ITSC), pp. 1456-1461, 2013.

[64] P. B. Sujit, S. Saripalli, and J. Borges Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles", Control Systems, IEEE, vol. 34, no 1, 42-59, 2014.

[65] R. Wallace, A. Stentz, C. E. Thorpe, H. Maravec, W. Whittaker, and T. Kanade, "First Results in Robot Road-Following", In the International Joint Conference on Artificial Intelligence, pp. 1089-1095, 1985.

[66] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, et al., "Stanley: The robot that won the DARPA Grand Challenge". In The 2005 DARPA Grand Challenge, Springer Berlin Heidelberg, pp. 1-43, 2007.

[67] J. Leonard, John, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, et al. "A perception-driven autonomous urban vehicle." Journal of Field Robotics, vol. 25, no 10, pp. 727-774, 2008.

[68] M. Buehler, K. Iagnemma, and S. Singh, eds. "The DARPA urban challenge: autonomous vehicles in city traffic". vol. 56. Springer, 2009.

[69] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, et al. "Odin: Team victorTango's entry in the DARPA urban challenge." Journal of Field Robotics, vol. 25, no 8, pp. 467-492, 2008.

[70] J. Snider, "Automatic steering methods for autonomous automobile path tracking." Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08, 2009.

[71] D. Poole, A. K. Mackworth, and R. Goebel. "Computational intelligence: a logical approach", New York: Oxford University Press, vol. 1, 1998.

[72] C. Castelfranchi. "Guarantees for autonomy in cognitive agent architecture." Intelligent agents. Springer Berlin Heidelberg, pp. 56-70, 1994.

[73] B. Burmeister, H. Afsaneh, and M. Guido, "Application of multiagent systems in traffic and transportation," Software Engineering, vol. 144, pp. 51-60, 1997.

[74] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles." Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on. IEEE, 2008.

[75] P. Patron, E. Miguelanez, J. Cartwright, and Y. R. Petillot, "Semantic knowledge-based representation for improving situation awareness in service oriented agents of autonomous underwater vehicles". In IEEE OCEANS, pp. 1-9, 2008.

[76] D. Ejigu, M. Scuturici, and L. Brunie, "An ontology-based approach to context modeling and reasoning in pervasive computing." Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on. IEEE, pp. 14-19, 2007.

[77] H. Chen, T. Finin, and A. Joshi. "An ontology for context-aware pervasive computing environments." The Knowledge Engineering Review, vol. 18, no 3, pp. 197-207, 2003.

[78] R. Wannous, J. Malki, A. Bouju, and C. Vincent. "Time integration in semantic trajectories using an ontological modelling approach." In New Trends in Databases and Information Systems, Springer Berlin Heidelberg, pp. 187-198, 2013.

[79] H. Chen, F. Perich, T. Finin, and A. Joshi. "Soupa: Standard ontology for ubiquitous and pervasive applications." In Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on, IEEE, pp. 258-267, 2004.

[80] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty. "Intelligent agents meet the semantic web in smart spaces." Internet Computing, IEEE, vol. 8, no. 6, pp. 69-79, 2004.

[81] M. N. Huhns and L. M. Stephens, "Multiagent Systems and Societies of Agents," in Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence, G. Weiss, Ed., ed: The MIT Press, pp. 79-120, 1999.

[82] M. Wooldridge and N. Jennings, “Agent theories, architectures, and languages: a survey,” Intelligent agents, pp. 1-39, 1995.

[83] M. Wooldridge, “Intelligent Agents,” in Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence, G. Weiss, Ed., ed: The MIT Press, pp. 25-78, 1999.

[84] R. A. Brooks, “A robust layered control system for a mobile robot.” Robotics and Automation, IEEE Journal of, vol. 2, no 1, pp. 14-23, 1986.

[85] P. Ögren. “Formations and obstacle avoidance in mobile robot control.”, Royal Institute of Technology, Stocholm, 2003.

[86] L. Zhao, R. Ichise, T. Yoshikawa, T. Naito, T. Kakinami, and Y. Sasaki, “Ontology-based decision making on uncontrolled intersections and narrow roads.” Intelligent Vehicles Symposium (IV), IEEE, pp.83-88, 2015.

[87] J. S. Albus, “4D/RCS: a reference model architecture for intelligent unmanned ground vehicles”, in AeroSense 2002. International Society for Optics and Photonics, pp. 303-310, 2002.

[88] M. Fiosins, J. Fiosina, J. P. Müller, and J. Görmer, “Agent-Based Integrated Decision Making for Autonomous Vehicles in Urban Traffic,” Advances on Practical Applications of Agents and Multiagent Systems pp. 173-178, 2011.

[89] J. Wahle, A. L. C. Bazzan, F. Klügl, and M. Schreckenberg, “Anticipatory Traffic Forecast Using Multi-Agent Techniques,” in Traffic and Granular Flow ’99, ed: Springer Berlin Heidelberg, pp. 87-92, 2000.

[90] B. Chen and H. H. Cheng, “A Review of the Applications of Agent Technology in Traffic and Transportation Systems,” Intelligent Transportation Systems, IEEE Transactions on, vol. 11, no 2, pp. 485-497, 2010.

[91] A. M. García-Serrano, D. T. Vioque, F. Carbone, and V. D. Méndez, “FIPA-compliant MAS development for road traffic management with a Knowledge-Based approach: the TRACK-R agents,” in Proceedings of Challenges in Open Agent Systems, Melbourne, Australia, 2003.

[92] V. R. Tomás and L. A. García, “Agent-Based Management of Non Urban Road Meteorological Incidents,” Multi-Agent Systems and Applications IV, pp. 213-222, 2005.

[93] B. Chen, H. H. Cheng, and J. Palen, “Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems,” Transportation Research Part C: Emerging Technologies, vol. 17, no. 1, pp. 1-10, 2009.

[94] M. Düring, and P. Pascheka. “Cooperative decentralized decision making for conflict resolution among autonomous agents.” Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on, pp. 154-161, 2014.

[95] C.-C. Wang, J. Healey, and M. Zhao, “Augmenting On-Road Perception: Enabling Smart and Social Driving with Sensor Fusion and Cooperative Localization,” presented at the 3rd Augmented Human International Conference, Megève, France, p. 21, 2012.

[96] S-W. Yang, and C-C. Wang. “Simultaneous egomotion estimation, segmentation, and moving object detection.” Journal of Field Robotics, vol. 28, no 4, pp. 565-588, 2011.

[97] Y. Dimopoulos and P. Moraitis, “Multi-Agent Coordination and Cooperation through Classical Planning,” presented at the Intelligent Agent Technology, pp. 398-402, 2006.

[98] H. Kautz, D. McAllester, and B. Selman, “Encoding Plans in Propositional Logic,” in Principles of Knowledge Representation and Reasoning, pp. 374-384, 1996.

[99] R. Olfati-Saber, A. Fax, and R. M. Murray. “Consensus and cooperation in networked multi-agent systems”, Proceedings of the IEEE, vol. 95, no 1, pp. 215-233, 2007.

[100] H. Qu, K. Xing, and T. Alexander, “An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots,” Neurocomputing, vol. 120, pp. 509-517, 2013.

[101] R. Claes, T. Holvoet, and J. V. Gompel, “Coordination in hierarchical pickup and delivery problems using delegate multi-agent systems,” in Workshop on Artificial Transportation Systems and Simulation, pp. 1-7, 2010

[102] F. Jiménez, J. Naranjo, O. Gomez and J. Anaya, “Vehicle Tracking for an Evasive Manoeuvres Assistant Using Low-Cost Ultrasonic Sensors.”, Sensors, vol. 14, no 12, pp. 22689-22705, 2014.

[103] F. Jiménez, J. Naranjo and O. Gómez. “Autonomous manoeuvring systems for collision avoidance on single carriageway roads.” Sensors, vol. 12, no 12, 16498-16521, 2012.

[104] Lee, Woon-Sung, Ji-Yong Lee, and Sang-Soo Park. "A New Approach to Forward Collision Avoidance." 3rd International Conference on Road Safety and Simulation. 2011.

[105] Milanés, Vicente, et al. "A fuzzy aid rear-end collision warning/avoidance system." *Expert Systems with Applications* 39.10 (2012): 9097-9107.

[106] C. Ackermann, R. Isermann, S. Min, and C. Kim, “Collision avoidance with automatic braking and swerving”, In World Congress, vol. 19, no 1, pp. 10694-10699, 2014.

[107] J.E. Naranjo, C. Gonzalez, R. García & T. De Pedro, “Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver”, IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 438-450, 2008.

[108] L. Guo, G. Ping-Shu, Y. Ming, Z. Yi-Bing, “Lane changing trajectory planning and tracking controller design for intelligent vehicle running on curved road”, Mathematical Problems in Engineering 2014, 2014.

[109] J. Lee, B. B. Park, K. Malakorn, and J. J. So, “Sustainability assessments of cooperative vehicle intersection control at an urban corridor,” Transportation Research Part C: Emerging Technologies, vol. 32, pp. 193-206, 2013.

[110] A. Abbas-Turki, F. Perronnet, J. Buisson, A. El-Moudni, M. Ahmane, and R. Zéo, “Cooperative intersections for emerging mobility systems,” presented at the 15th meeting of the Euro Working Group on transportation, Marne La Valle, Paris, 2012.

[111] M. Ahmane, A. Abbas-Turki, F. Perronnet, J. Wub, A. E. Moudni, J. Buisson, et al., “Modeling and controlling an isolated urban intersection based on cooperative vehicles,” Transportation Research, vol. 28, pp. 44-62, 2013.

[112] M. Maile, F. Ahmed-Zaid, C. Basnyake, L. Caminiti, S. Kass, M. Losh, J. Lundberg, D. Masselink, E. McGlohon, P. Mudalige, C. Pall, M. Peredo, Z. Popovic, J. Stinnett, and S. VanSickle, “Cooperative Intersection Collision Avoidance System Limited to Stop Sign and

Traffic Signal Violations (CICAS-V)", RITA Intelligent Transportation Systems Joint Program Office, No. Final Report-Phase 1, 2008.

[113] D. Caveney, W. B. Dunbar, "Cooperative Driving: Beyond V2V as an ADAS sensor", Intelligent Vehicles Symposium, Spain, pp. 529-534, 2012.

[114] D. Sun, A. Kleiner, and B. Nebel. "Behavior-based multi-robot collision avoidance." Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, pp. 1668-1673, 2014.

[115] J. Huang, and H. Tan, "Design and implementation of a cooperative collision warning system", In Intelligent Transportation Systems Conference (ITSC'06), IEEE, pp. 1017-1022, 2006.

[116] A. Fujimori, M. Teramoto, P. N. Nikiforuk, M. M. Gupta, "Cooperative collision avoidance between multiple mobile robots." Journal of Robotic Systems, vol. 17, no 7, pp. 347-363, 2000.

[117] Van den Berg, S. Guy, D. Manocha, "Reciprocal n-body collision avoidance." Robotics research, Springer Berlin Heidelberg, 3-19, 2011.

[118] J. Alonso-Mora, A. Breitenmoser, P. Beardsley and R. Siegwart. "Reciprocal collision avoidance for multiple car-like robots." Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, pp. 360-366, 2012.

[119] R. Eigner, and G. Lutz. "Collision avoidance in vanets-an application for ontological context models." Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on. IEEE, pp. 412-416, 2008.

[120] CH. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I", IEEE Transactions on systems, man, and cybernetics, vol. 20, no. 2, pp. 404-418, 1990.

[121] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", IEEE Transactions on Systems, man and cybernetics, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[122] W.-K. Chen, "Linear Networks and Systems", Belmont, CA: Wadsworth, pp. 123–135, 1993

[123] M. Bradley, "Path planning and path following for an autonomous car". Master of Science, Middle Tennessee State University, 2012.

[124] G. Genta, "Motor vehicle dynamics: modeling and simulation", World Scientific, vol. 43, 1997.

[125] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban Mobility", International Journal On Advances in Systems and Measurements, vol. 5, pp. 128-138, 2012.

[126] G. Colmenares, and I. Benyahia, "Cooperative Driving Experiments Based on a 3D Components Environment.", IFAC-PapersOnLine, vol. 48, no. 3, pp. 2351-2355, 2015.

[127] Dréo, J., Pétrowski, A., Siarry, P. and Taillard, E., "Métaheuristiques pour l'optimisation difficile", Eyrolles, 2003.

[128] G. Colmenares, I. Benyahia, M. Shawky, and M. Zaremba. "Intelligent Agents for Cooperative Driving: Challenges and Solutions," presented at the 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2015), Ottawa, Canada, pp.76-84, 2015.

[129] G. Colmenares, and I. Benyahia. "Role of Vehicle’s Intention on the Optimization of Collision Avoidance Strategies for Cooperative Driving.", in 24th Annual ITS World Congress 2017, Montreal, Canada, 2017.

[130] Colmenares, G. "Comportement de véhicules autonomes pour éviter des collisions et minimiser les incidents des véhicules d’urgence", presented at the Semaine de la recherche, Université du Québec en Outaouais, Gatineau, Canada, 2015.

Appendix A

Basic concepts on vehicle kinematics

Kinematic model

With a kinematic model it is possible to the study and evaluate the dynamics of a vehicle[1], they could include lateral and forward displacement; also, based on this model, the rotation center can be easily computed by kinematic means. Possible translations and rotations in the different axis of three dimensions, give to the vehicle six degrees of freedom: roll, pitch and yaw, are rotations in the x-axis, y-axis and z-axis, respectively[2]; also, two forces can be considered in a vehicle model: longitudinal and lateral.

Modeling the steering control of a motor vehicle is commonly based on the Ackerman model[3], which is a geometrical solution to avoid tires from slipping sideways while performing a curve. All wheels are set to turn around a common central point; this means that each wheel has its own turning angle (Fig. A.1).

---

[1] M. A. Sotelo, "Lateral control strategy for autonomous steering of Ackerman-like vehicles", Robotics and Autonomous Systems, vol. 45, no 3, pp. 223-233, 2003.

[2] P. Gáspár, "Design of integrated control for road vehicles." Robust Control and Linear Parameter Varying Approaches. Springer Berlin Heidelberg, pp. 213-235, 2013.

[3] H. R. Everett, "Sensors for mobile robots: theory and application". AK Peters, Ltd., 1995.

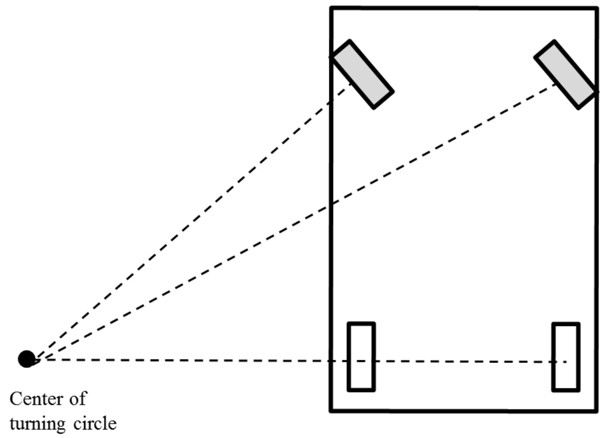

Figure A.1 Basic Ackerman model

To simplify calculations, Ackerman et al.[4] defined a two-wheeled approximation of the original model, known as the classical single-track model or bicycle model; it is widely used as main kinematic model [20], [105], [1]. This model assumes that the two front wheels are collapsed to the center of the axle, and the same for the rear wheels; it is depicted in Figure A.2.

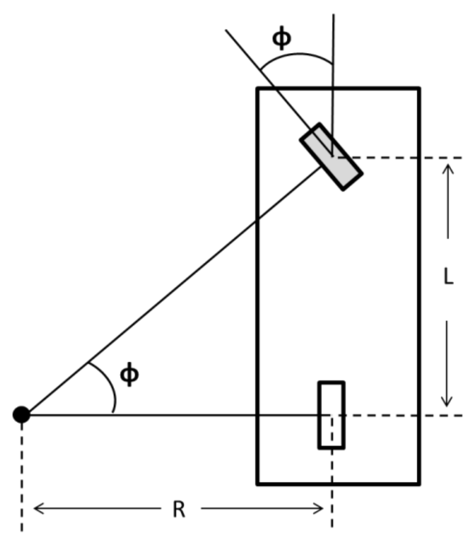

Figure A.2. Single track model [4]

---

[4] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser, and V. Utkin, "Linear and nonlinear controller design for robust automatic steering", Control Systems Technology, IEEE Transactions on, vol. 3, no 1, 132-143, 1995.

Where ϕ is the steering angle, R is the radius of the circumference and L is the wheelbase of the vehicle.

Appendix B

Description of attributes of vehicles and other objects

This appendix presents a description of the attributes of vehicles and other objects that can be present in a collision scenario while being simulated in our visualization environment. Each vehicle in a scenario has several properties that have to be set before starting the simulation. The variables used to set the initial state of the vehicle, as well as some constraints are presented in Table B.1.

Table B.1 Vehicle setup variables

Variable	Description
Id	Used to identify and control the vehicle during the simulation.
Type	Type of the vehicle.
Max speed	Max speed of the vehicle.
Location	Initial coordinates of the vehicle in the map.
Orientation	Initial orientation of the vehicle in the map. In degrees, relative to a global common set of coordinates.
Connected	Indicates if the vehicle has communication capabilities.
Antenna range	If it is a connected vehicle, this value indicates the radius of the antenna range. The vehicle can communicate with connected vehicles or equipped infrastructure within the range.
Rerouting capabilities	Indicates if the vehicle can recalculate its route.
Routing algorithm	Indicates which of the implemented routing algorithms this vehicle uses.

The kinematics of the vehicle are controlled by the bicycle model, presented in appendix A, since it is a standard used by the majority of researches in the field. Attributes used to control the vehicle on the simulated roads, to detect the possibility of collision, as well as physical constraints are described in Table B.2.

Table B.2 Vehicle kinematics attributes

Attribute	Description
Length	Length of the vehicle, in meters.
Width	Width of the vehicle, in meters.
Min turning radius	The radius of the circle drawn by the car at maximal steering angle.
Wheel base	Distance from the center of the front wheels to the center of the rear wheels.
Max engine force	The maximal force applicable by the engine, in Newtons.
Max brake force	The maximal force applicable by the brakes, in Newtons.
Frontal area	Area of the front of the car, used in combination with the drag coefficient and density of the air to compute the drag coefficient of the car while moving.
Roll resistance	Energy lost when the tires are rolling.
Steer speed	The speed at which the steering wheel moves, in radians.
Wheels position	A two coordinates vector for each wheel. Position is relative to the center of the car, in meters.

Other attributes are used by the physics engine in order to compute collisions and to produce a realistic behavior of 3D objects in the simulation. Such constants are presented in Table B.3.

Table B.3 Additional attributes used by the physics engine

Attribute
Chassis density
Coefficient of friction
Coefficient of restitution
Linear damping
Angular damping
Drag coefficient
Density of the air
Rolling resistance

Other objects that can be simulated in the 3D environment are traffic lights, traffic signs, streets, lanes and people; Table B.4 describes the attributes used to configure those objects in the simulation.

Table B.4 Simulation objects and their descriptions

Object	Description
Traffic light and signs	
Id	Used to identify and access variables of the object in the 3D environment.
Location	Current coordinates of the object in the map.
Orientation	Current orientation. In degrees, relative to a global common set of coordinates.
Equipped	Indicates if the object has communication capabilities.
Street	
Id	Used to identify and access variables of the street in the 3D environment.
Location	Coordinates of the street in the map.
Orientation	Current orientation. In degrees, relative to a global common set of coordinates.
List of lanes	Array of lanes composing this street.

Next streets	Array of streets to which this street leads.
Lane	
Id	Used to identify and access variables of the lane in the 3D environment.
Location	Coordinates of the lane in the map.
Orientation	Current orientation. In degrees, relative to a global common set of coordinates.
Parent street	Id of the street in which this lane is.
Connected lanes	Array of lane ids to which this lane is connected.
People	
Id	Used to identify persons simulated in the 3D environment.
Location	Current coordinates of the person in the map.
Orientation	Current orientation. In degrees, relative to a global common set of coordinates.

Appendix C
Force, acceleration and heading angle computation

In this appendix, we explain the formulae needed to compute the force needed on the vehicle's engine in order to attain certain expected speed. As one of the objectives of this research is to produce realistic results, it is essential to consider the variables that rule the displacement of the vehicle over the roads, such as traction, drag and roll. So, from the work of Genta [118], we know that:

$$F_{traction} = u \cdot EngineForce \tag{C.1}$$

Where *u* is a unit vector that indicates the vehicle's heading, and *EngineForce* is a percentage of the max engine force depending on the current and expected speeds' difference.

$$F_{drag} = -\frac{1}{2} \cdot \rho \cdot s^2 \cdot C_D \cdot A \tag{C.2}$$

Where ρ is the density of air, *s* is the vehicle's speed, A is the frontal area of the car and C_D is the drag coefficient.

$$F_{roll} = -C_r \cdot v \tag{C.3}$$

Where C_r is the rolling resistance constant and *v* is the velocity vector. For clarification purposes, the vehicle's velocity is a vector composed of the magnitude of the speed and a direction; then, the speed is a scalar. Subsequently, the longitudinal force applied to the vehicle when it has positive acceleration is:

$$F_{long} = F_{traction} + F_{drag} + F_{roll} \tag{C.4}$$

Notice that formula (4.4) is valid when current speed is lower than expected speed, i.e.: the vehicle has to accelerate to get to the expected speed. On the contrary, if current vehicle's speed is higher than expected, then it needs to decelerate; therefore, there is a braking force, which is oriented in the opposite direction:

$$F_{brake} = -u \cdot C_{braking} \tag{C.5}$$

Where $C_{braking}$ is a percentage of the max brake force depending on the current and expected speeds' difference. Hence:

$$F_{long} = F_{brake} + F_{drag} + F_{roll} \tag{C.6}$$

Although the physics engine computes acceleration, speed and location of objects in the environment, for collision prediction purposes we need to compute possible values of these variables, given different vehicle avoidance strategies, without actually moving the vehicle on the simulation. So, from [118] we know that, given the force, acceleration can be computed by:

$$a = \frac{F}{M} \tag{C.7}$$

Where F is the net force on the vehicle (in Newtons) and M is the mass. As acceleration is the change of velocity with respect to time, then the velocity can be computed by:

$$v = v + dt \cdot a \tag{C.8}$$

Where dt is the time step of the simulation; in other words, is the time that passes between one calculation of the vehicle's velocity and the next. Finally, given the velocity it is possible to compute the vehicle's location using:

$$p = p + dt \cdot v \tag{C.9}$$

While the vehicle has a steering angle different from zero, its orientation (heading direction) has to be updated. To do so, we need to compute the angular velocity, which is the rate at which the vehicle turns; it is expressed in radians per second and it is given by:

$$\omega = \frac{s}{R} \tag{C.10}$$

Where s is the speed and R is the radius of the circle that is being drawn by the vehicle with current steering angle. To compute R, we have to go again to the Ackerman single track model (Figure 5.2), where L is the distance between wheel axles (also known as wheel base), Φ is the steering angle, and R is the radius of the circle.

From there, with simple geometry we have:

$$sin(\Phi) = \frac{L}{R} \tag{C.11}$$

Therefore, the value of R is:

$$R = \frac{L}{sin(\Phi)} \tag{C.12}$$

Finally, the vehicle's heading h at each time step is given by:

$$h = h + dt \cdot \omega \tag{C.13}$$