UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Cooperative driver assistance system for the lane change

Andrés Eduardo Gómez Hernandez

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Andrés Eduardo Gómez Hernandez

Cooperative driver assistance system for the lane change

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
March 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

G633c Gómez Hernandez, Andrés Eduardo
Cooperative driver assistance system for the
lane change / Andrés Eduardo Gómez Hernandez;
orientador Denis Fernando Wolf. -- São Carlos,
2018.
139 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2018.

1. Driver Assistance Systems. 2. Cooperative
Intelligent Transport Systems. 3. Driving
simulator. 4. Bayesian Networks. 5. Vehicular
communications. I. Wolf, Denis Fernando , orient.
II. Título.

Andrés Eduardo Gómez Hernandez

**Sistema cooperativo de assistência ao motorista para a
mudança de faixa**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
Março de 2018

*Dedico este trabalho a Deus, por seu infinito amor e bondade, e por manter sempre firme suas
mãos nos momentos em que mais precisei dele,
a minha amada esposa, Ayumi, pela força que dá em minha vida, e por ser o sol nascente de
nossa família,
a minha mãe, Nelly, por me ensinar que agindo corretamente, coisas boas sempre chegam,
a minha família, por ser uma das motivações de minha vida,
ao generoso povo Brasileiro "gigante pela própria natureza", por ampliar minha visão da
ciência e do mundo.*

ACKNOWLEDGEMENTS

Agradeço ao generoso povo Brasileiro, através da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (*Capes*), que me ajudou a cumprir um de meus sonhos nesta vida. Espero que este enorme gesto de altruísmo por um irmão Colombiano seja retribuído através de uma sociedade que a cada dia se torne mais próspera, unida e equitativa. Gostaria também de agradecer à *Universidade de São Paulo*, ao *Instituto de Ciências Matemáticas e de Computação* e ao *Laboratório de Robótica Móvel*, por todas as oportunidades que me ofereceram neste tempo de estudos.

Agradeço ao professor Denis por possibilitar esta valiosa oportunidade. Aos professores Arthur de Miranda neto, Victor Etgens, Sebastien Glaser e Yasser Alayli por me permitirem conhecer a visão Francesa da ciência. À Joanna Sabbatini, minha editora de inglês, quem, com suas observações claras e pontuais, me ajudou a melhorar o texto de minha tese. Finalmente, agradeço especialmente a todas as pessoas que me ajudaram nesta grande aventura de minha vida.

“Santandereanos siempre adelante, Santandereanos ni un paso atrás.”
(Himo de Santander-Colombia)

ABSTRACT

ANDRÉS EDUARDO, G. H. **Cooperative driver assistance system for the lane change**. 2018. 139 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

The increase in the number of deaths due to ground traffic accidents is a global problem. In such context, the development of new vehicular technologies is considered an alternative to improve road safety. Within the field of new vehicle technologies, it is possible to find driver assistance systems. These systems interact in an active or passive way with the driver, reducing their workload by presenting information about their surroundings, which may imply the safer direction of a land vehicle. Taking into account that one of the main reasons for road traffic fatalities in the world is the lane change in a road, hereby we created a cooperative driver assistance system for the lane change, arising from the combination of a kinematic model and a probabilistic graphical one. By combining these two models, we try to improve the response in the assistance of the system, given the direct dependence of the system with a human. Due to the latter, the response of such systems cannot be deterministic in nature. One of the motivations to use probabilistic graphical models is the flexibility of this machine's learning technique in modeling the problem addressed in this thesis. In addition to this contribution of applying a specific probabilistic graphical model in our assistance system, other contributions can be found in this thesis, including the development of a Driving simulation platform with a reconfigurable structure. The ability to reconfigure the structure of the driving simulator platform was of great importance for the development and evaluation of the assistance system hereby proposed in each of its stages. In addition, the decision to model a cooperative approach in our assistance system is due to the great potential of the vehicular communications with respect to improving transport safety and efficiency. The moderate cost that is being projected in vehicular communications is another relevant fact. Finally, the description and application of our assistance system model can be considered as a possibility in the area for the development of an application that needs a close response to the reality, based on the uncertainties present in the problem under consideration.

Keywords: Driver Assistance Systems, Cooperative Intelligent Transport Systems, Driving simulator, Bayesian networks.

RESUMO

ANDRÉS EDUARDO, G. H. **Sistema cooperativo de assistência ao motorista para a mudança de faixa**. 2018. 139 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

O aumento no número de mortes por causa de acidentes de tráfego terrestre é um problema global. No âmbito dessa problemática, o desenvolvimento de novas tecnologias veiculares é considerado uma alternativa para melhorar a segurança viária. Dentro do domínio das novas tecnologias veiculares, é possível encontrar sistemas de assistência ao motorista. Esses sistemas interagem de maneira ativa ou passiva com o motorista, conseguindo reduzir sua carga de trabalho, apresentando informações de seu entorno, o que pode implicar uma direção mais segura de um veículo terrestre. Levando em consideração que um dos principais motivos de mortes por acidentes de tráfego terrestre no mundo é a mudança de faixa em uma pista, neste trabalho, realizamos a tarefa de criar um sistema cooperativo de assistência ao motorista para a mudança de faixa, a partir da combinação de um modelo cinemático e de um modelo gráfico probabilístico. Mediante a combinação destes dois modelos, procuramos melhorar a resposta na assistência por parte do sistema, dada a dependência direta que o sistema tem dos humanos. Por essa última razão, a resposta deste tipo de sistemas não pode ser determinística por natureza. Uma das motivações para utilizar os modelos gráficos probabilísticos é a flexibilidade da técnica de machine learning em modelar o problema abordado nesta tese. Além dessa contribuição de aplicar um modelo gráfico probabilístico específico em nosso sistema de assistência, outras contribuições podem ser encontradas nesta tese, incluindo o desenvolvimento de uma plataforma de simulação para a condução, com uma estrutura reconfigurável. A capacidade de poder reconfigurar a estrutura da plataforma de simulação foi de grande importância para o desenvolvimento e avaliação do sistema de assistência proposto nesta tese, em cada uma de suas etapas. Além disso, a decisão de modelar um enfoque cooperativo, em nosso sistema de assistência, se deve ao grande potencial que tem as comunicações veiculares com respeito à melhora da segurança e da eficiência do transporte. O custo moderado que está sendo projetado nas comunicações veiculares é outro fato relevante. A descrição e aplicação de nosso modelo final podem ser considerados mais uma possibilidade na área para o desenvolvimento de uma aplicação, que precise de uma resposta próxima da realidade, a partir das incertezas presentes no problema considerado.

Palavras-chave: Sistema de assistência à condução, Sistemas de transporte inteligentes cooperativos, simulador de condução, Redes Bayesianas.

LIST OF FIGURES

Figure 1 – Top ten causes of death among people aged 15–29 years, 2012	29
Figure 2 – Type versus severity of accidents on federal highways (2014) (In %)	30
Figure 3 – Road fatalities in the EU since 2001.	31
Figure 4 – Selection of the best <i>test track</i> for an application, in function of the realism and the control.	47
Figure 5 – The 3D simulation model of CaRINA 2.	50
Figure 6 – Simulation platform architecture example.	51
Figure 7 – Driving simulation platform configuration used in the cooperative vehicle system.	55
Figure 8 – Leader vehicle Velocity (<i>L, black</i>), Follower vehicle velocity (<i>F, red</i>), velocities variation (<i>L-F, blue</i>), communication delay 0.01 s.	58
Figure 9 – Leader vehicle Velocity (<i>L, black</i>), Follower vehicle velocity (<i>F, red</i>), velocities variation (<i>L-F, blue</i>), communication delay 1.0 s.	58
Figure 10 – Velocity and distance between the vehicles with communication delay 0.01 s.	60
Figure 11 – Velocity and distance between the vehicles with communication delay 0.333 s.	60
Figure 12 – Vehicles with the same direction.	64
Figure 13 – Vehicles with opposite direction.	67
Figure 14 – System design	69
Figure 15 – Result of <i>DKF</i> application in the <i>Ev</i> displacement	70
Figure 16 – Results of d_a measures for each collision situation	74
Figure 17 – Example of a Bayesian network.	78
Figure 18 – Diagram showing a model where time slices are used to represent a temporal process.	82
Figure 19 – A 2-time slice Bayesian network (2 – <i>TBN</i>), which shows the dynamic and observation models using two consecutive time slices of a process	83
Figure 20 – Decision's states	90
Figure 21 – 2 – <i>TBN</i> model structure	95
Figure 22 – New system design	99
Figure 23 – Definition of lanes in the simulation scenario.	101
Figure 24 – <i>Collision situation 1</i> : vehicles making (and not making) a lane change maneuver with opposite direction. class 1 : warning; class 2 : change; class 3 : same.	102

Figure 25 – <i>Collision situation 2: vehicles on two-way road. class 1: warning; class 2: change; class 3: same.</i>	103
Figure 26 – <i>Collision situation 3: following vehicle making a lane change maneuver with traffic on the two lanes. class 1: warning; class 2: change; class 3: same.</i>	104
Figure 27 – Flowchart of the proposed method.	108
Figure 28 – Image input.	109
Figure 29 – Image processing.	110
Figure 30 – Traffic light detection.	111
Figure 31 – A Markov chain with 4 states.	112
Figure 32 – Driving simulation platform configuration used to evaluate the <i>DAS'</i> influence in a driver.	116
Figure 33 – Hardware used in the influence evaluation of a Driver assistance system in a driver.	118
Figure 34 – Survey questions	120
Figure 35 – A participant's log without <i>DAS'</i> model.	123
Figure 36 – A participant's log with <i>DAS'</i> model.	123

LIST OF CHARTS

Chart 1 – Random variable Lane (L)	92
Chart 2 – Random variable Vehicle-direction (Vd)	93
Chart 3 – Random variable Location (Lc)	93
Chart 4 – hypothesis variable Decision (D)	94
Chart 5 – Comparison between the two versions of the driving simulation platform . .	97

LIST OF ALGORITHMS

Algoritmo 1 – Collision Warning	63
Algoritmo 2 – Lane change	87
Algoritmo 3 – Critical	92
Algoritmo 4 – Inference	96
Algoritmo 5 – Following case	115

LIST OF TABLES

Table 1 – Statistics features of L-F velocity variations due to delay times in communication.	59
Table 2 – Zones around Ev	66
Table 3 – Comparison between the signals from figure 15	70
Table 4 – Range detection distance of the warning signals by zone around Ev	75
Table 5 – FI score for the collision situation 1	105
Table 6 – FI score for the collision situation 2	105
Table 7 – FI score for the collision situation 3	105
Table 8 – State transition probabilities.	113
Table 9 – The observation probabilities.	113
Table 10 – Confusion matrix of results without HMM	115
Table 11 – Confusion matrix of results with HMM	115
Table 12 – Results group without DAS	122
Table 13 – Results group with DAS	122

LIST OF ABBREVIATIONS AND ACRONYMS

2 – <i>TBN</i>	2-time slice Bayesian network
ADAS	Advanced Driver Assistance Systems
AV	Autonomuos Vehicle
BN	Bayesian Networks
BSM	Basic safety message
C-ITS	Cooperative Intelligent Transport Systems
CACC	Cooperative Adaptive Cruise Control
CPD	Conditional Probability Distribution
CW	Collision Warning
DAG	Directed Acyclic Graph
DAS	Driver Assistance Systems
DBN	Dynamic Bayesian Network
EU	European Union
Ev	Ego vehicle
HMM	Hidden Markov Model
LRM	Mobile Robotics Laboratory
LTA	Land Traffic Accidents
MAP	Maximum a Posteriori Assignment
MORSE	Modular Open Robots Simulation Engine
NHTSA	National Highway Traffic Safety Administration
NIC	Network Interface Communication
Ov	Other vehicle
PGM	Probabilistic Graphical Models
PID	Proportional Integral Derivative
ROS	Robot Operating System
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Overall
VANET	Vehicular Ad Hoc Network
VDM	Vehicular Dynamic Model

CONTENTS

1	INTRODUCTION	29
1.1	Problem statement and challenges	31
1.2	Contributions	32
1.3	Thesis outline	32
2	RELATED WORKS	34
2.1	Initial Considerations	34
2.1.1	<i>Simulation platforms</i>	34
2.1.2	<i>Simulated systems</i>	35
2.1.2.1	<i>Vehicular communications</i>	38
2.1.3	<i>Non-simulated systems</i>	39
2.2	Final considerations	43
3	DRIVING SIMULATION PLATFORM	44
3.1	History	45
3.2	Classification of driving simulators	45
3.2.1	<i>Low-cost driving simulators</i>	46
3.2.2	<i>Medium-cost driving simulators</i>	46
3.2.3	<i>High-cost driving simulators</i>	46
3.3	Advantages and disadvantages of driving simulators	46
3.4	Validation of driving simulators	47
3.5	Driving simulators used by the automotive industry	47
3.6	Driving simulators used in researches	48
3.7	Driving simulation platform architecture	49
3.7.1	<i>Vehicular dynamic model</i>	51
3.8	Final considerations	52
4	COOPERATIVE INTELLIGENT TRANSPORT SYSTEMS	53
4.1	Simulation platform for cooperative vehicle systems	54
4.1.1	<i>Control</i>	54
4.1.2	<i>Driving simulation platform configuration</i>	55
4.1.3	<i>Results</i>	57
4.1.3.1	<i>Delay effect on the CACC system</i>	57
4.1.3.2	<i>CACC system validation</i>	59

4.2	Cooperative collision warning for driving assistance	61
4.2.1	Model	61
4.2.1.1	<i>Vehicles in the same direction</i>	63
4.2.1.2	<i>Location of Ov around Ev</i>	65
4.2.1.3	<i>Vehicles in opposite directions</i>	66
4.2.2	System design	67
4.2.2.1	<i>V2V communications</i>	67
4.2.2.2	<i>Vehicle simulation</i>	68
4.2.2.3	<i>Vehicle position estimation</i>	68
4.2.3	Simulation results	70
4.2.3.1	<i>Experiments</i>	71
4.2.3.2	<i>Results</i>	72
4.3	Final considerations	76
5	APPLICATION OF BAYESIAN NETWORKS ON COOPERATIVE DRIVING ASSISTANCE SYSTEMS	77
5.1	Bayesian Networks	78
5.1.1	Building models	79
5.1.2	Inference	81
5.2	Dynamical Bayesian Networks	81
5.2.1	Building models	82
5.2.2	Inference	83
5.3	Driver assistance system for lane change	83
5.4	Cooperative driver assistance system for lane change	86
5.4.1	Kinematic model	86
5.4.2	Probabilistic graphical model	89
5.4.2.1	<i>Obtaining and calculating parameters</i>	90
5.4.2.2	<i>Storage of parameter values obtained</i>	91
5.4.2.3	<i>Determination of the random variables states</i>	92
5.4.2.4	<i>Learning the 2-TBN model structure</i>	93
5.4.2.5	<i>Learning of the 2-TBN parameters</i>	96
5.4.2.6	<i>Inference in the 2-TBN model</i>	96
5.4.3	System design	97
5.4.4	Simulation results	98
5.5	Final considerations	106
6	OTHER CONTRIBUTIONS	107
6.1	Traffic lights detection and state estimation using hidden markov models	107
6.1.1	Method	107

6.1.1.1	<i>Image input and image processing</i>	108
6.1.1.2	<i>Traffic lights detection</i>	110
6.1.1.3	<i>State detection</i>	111
6.1.1.4	<i>Estimation of state obtained and given state</i>	112
6.2	Influence evaluation of a driving assistance system on a driver	115
6.2.1	<i>Driving simulation platform configuration</i>	115
6.2.2	<i>Experiments</i>	117
6.2.2.1	<i>The lead vehicle stopped</i>	118
6.2.2.2	<i>A vehicle changing lanes in the same direction</i>	119
6.2.2.3	<i>Discussion</i>	119
6.3	Final considerations	124
7	CONCLUSIONS	125
7.1	Future work	127
	BIBLIOGRAPHY	129

INTRODUCTION

Public roads are necessary for the development of our societies. The clearest example is the importance of these infrastructures to the economy of any country. However, every time we use public roads around us, there is a risk of death or injury for all users. Estimates from the World Health Organization (2015), indicate that the age group with the highest number of road traffic fatalities is 15 to 29 years (Figure 1). In addition, the World Health Organization also reports that 90% of injury deaths occur in low and middle-income countries, which hold about 54% of vehicles worldwide. In the American continent, only two countries are considered to be high-income (Canada and United States of America), the others are considered to be middle-income countries.

Figure 1 – Top ten causes of death among people aged 15–29 years, 2012

Source: WHO (2015).

Considering the Pan American Health Organization (2015), the level of traffic accident risk depends on several physical and social factors, for example: *i*) country of residence; *ii*) age of the road user; *iii*) type of user; *iv*) rural or urban means; *v*) velocity; and *vi*) type of

vehicle, among others. With regard to these factors, the World Health Organization, based on the conclusions presented in (WHO, 2015), highlights the need for policy management by the nations that help put safer vehicles and roads. Currently, according to this same international body, vehicles sold in 80% of the world's countries do not meet minimum safety standards.

In Brazil, according to the Institute of Applied Economic Research (2015) (*IPEA*, by the acronym in Portuguese) on federal roads inspected by the Federal Highway Police (*PRF*, by the acronym in Portuguese), in 2014, there were 169,163 Land Traffic Accidents (LTA), of which 8,227 people lost their lives and about 100,000 were injured. Approximately 67% of accidents involving fatalities occurred in rural areas¹, and 23% of the deaths were caused by speeding or undue overtaking (which includes, in our understanding, the lane change on a road). In the same study, *IPEA* highlights that among the accidents on federal highways, the front collision was responsible for 33,7% of the deaths, which corresponds to 40.4 people per hundred accidents. Figure 2 shows the percentages of other types of accidents on Brazilian highways.

Figure 2 – Type versus severity of accidents on federal highways (2014) (In %)

Source: *IPEA* and *PRF* (2015).

IPEA also made an estimate on the cost of traffic accidents to the Brazilian society. In this study, *IPEA* reports that road accidents in the country cost 40 billion Reais, while in urban areas the cost exceeds 10 billion Reais. From this perspective, *Ladeira et al.* (2017), points out that in addition to the high mortality in the country, the *LTAs* have a strong impact on health services and also for the society in general. This idea is reinforced by *Andrade* (2015), who also considers that one of the consequences of *LTAs* is the permanent loss of people's motor capacities, which can lead to long-term inactivity.

In this worrying context, it is also important to note how effective intervention with policy and research has benefited the corresponding reduction in deaths and injuries by *LTA*. The European Union (EU) is an example of such collaborative work between research and policy

¹ Roads and highways (*LEGISLAÇÃO*, 2010).

measures. The Road fatalities in the *EU* since 2001 are shown in [Figure 3](#). In this figure, it is possible to verify the goal projected by the Europeans in the reduction of fatalities in the roads, as well as the results obtained until the previous year.

Figure 3 – Road fatalities in the EU since 2001.

Source: [EU \(2017c\)](#).

Among all measures taken by the European Union to achieve its goal, we highlight: *i)* the methodologies for licensing new drivers; *ii)* data collection of accidents; *iii)* the improvement of road infrastructure; and *iv)* the most important point for our academic context, the great investment in road safety research ([EU, 2017a](#); [EU, 2017b](#)). Within the field of projects developed by the European Union, one of the most significant is the development of vehicular technologies (i.e., active safety, passive safety, periodic technical inspection, vehicle emissions).

1.1 Problem statement and challenges

Within the development of vehicle technologies, there are Driver Assistance Systems (DAS). According to [Volkswagen \(2017\)](#), the *DAS* contribute to the detection of critical situations before they arise, and help reduce the risk of accidents. These systems make the driver's task more comfortable and convenient and they also reduce their workload.

Given the problem of accidents with fatalities due to frontal collisions in the Brazilian context, which are caused by speeding and undue overtaking, and considering the development of vehicle technologies (e.g. *DAS*), hereby we formulate the following problem: *Given a road environment, to determine when a driver can change lanes, through information presented to him. This information is intended to alert the driver to possible collisions between nearby vehicles by combining a kinematic and stochastic model. These models use information shared among vehicles, from a vehicular communications system.*

Vehicle communications have great potential for improving the safety and efficiency of land transportation (EU, 2016). This is possible due to the increase of available information that can be shared between vehicles. Another relevant aspect of the use of this technology is the moderate cost it has shown in its developmental stage (NHTSA, 2014). These aspects have turned vehicular communications into a strategic point of the Intelligent Transportation Systems in the European and United States agencies (EU, 2016; BARBARESSO *et al.*, 2014). According to the National Highway Traffic Safety Administration (NHTSA) (2017), vehicular communications technology has the potential to prevent 80% of multi-vehicle accidents.

Additionally, taking into account that the state of the art of the area emphasizes the importance of ensuring safe conditions for people in the tests (through resources and facilities suitable for their development), for the completion of the experiment proposed in this thesis, we decided to use a driving simulation platform.

1.2 Contributions

The main contributions presented in this thesis are as follows:

- The development of a Driving simulation platform, which allows it to adapt to the needs of the developed researches. This simulation platform was designed from a set of open-source tools, aiming at being freely distributed.
- The development of a cooperative deterministic model that can warn the driver of possible collisions against nearby vehicles. In this model, a new algorithm was used, which locates the vehicles around the Ego-vehicle in order to inform the driver.
- The proposal of a new cooperative driver assistance system for the lane change, in which the combination of the kinetic model developed in the previous contribution, and the probabilistic model 2-Time Bayesian Network (2 – TBN) was considered.

1.3 Thesis outline

The remanding of this thesis is divided into 6 chapters. In Chapter 2 we provide the relevant technical background to the reader, introducing the concepts and methods used throughout this thesis. In Chapter 3, we present all the aspects that make up the architecture of the driving simulation platform that was developed in this thesis. In Chapter 4, we used the first version of the simulation platform to develop a Cooperative Adaptive Cruise Control system. In addition, in the same chapter, through an improvement in the simulation platform communications, a new cooperative driver assistance system is developed. In Chapter 5, we present and justify each of the steps in the development of the new cooperative assistance system for the lane change. In this chapter, in addition to the presentation of the new improvements in the driving simulation

platform, a detailed explanation of the evaluations performed in the assistance system is also made. In [Chapter 6](#), we present further contributions made in the area during the doctoral period. Finally, in [Chapter 7](#) we draw the conclusions and suggest future works.

RELATED WORKS

2.1 Initial Considerations

To support the statement presented in subsection 1.1, a search was conducted within the *IEEE* and *Springer* databases for related works. It was done using three key topics: *i)* Ground Vehicles, *ii)* Collaborative or Cooperative systems, and *iii)* Driver assistance. A search string of synonyms for these key topics was also used in order to obtain a wider range of results:

(vehicle OR automobile OR transport OR vehicles OR automobiles OR transports) AND (land OR terrestrial OR ground) AND (collaborative OR cooperative) AND ("driver assistance" OR "assisted driving" OR "assistive driving" OR "assisted vehicle" OR "assistive vehicle" OR "supportive driving" OR "assisted vehicles" OR "assistive vehicles")

Time limits were not placed for the search and neither were filters for the titles, abstracts and keywords. In order to trim down the large amount of studies resulted from the search of both databases it was read the title, the abstract and the keywords of each, excluding those which did not have cooperative driving assistance in land vehicles as their main objective. The remaining studies were read in full and then divided in three topics for easier and faster reference: *i)* simulation platforms; *ii)* simulated systems; and *iii)* non-simulated systems. Through the reference lists of these selected studies it was possible to identify additional related works that were also used as reference.

2.1.1 *Simulation platforms*

simulation platforms can help decrease the complexity of developing and validating prototypes of Driver Assistance Systems (*DAS*). As an example, in a simulator it is possible to evaluate the prototypes' behavior under different controlled circumstances. Simulators are also a low-cost tool and resource. There are currently several simulation platforms available

(e.g., CarSim/TruckSim, rFactor, PreScan, TORCS, Delta3D, Groovesim, Pro-SiVIC, Ansible Motion's series simulators). Some of these platforms are commercial and some are open source. Below, we present two simulation environments developed in the area of robotics that can help in the development of *DAS*.

Gazebo (KOENIG; HOWARD, 2004) was designed to recreate a dynamic-multi-robot 3D environment filled with the most complex scenarios for the next generation of mobile robots. *Gazebo's* realistic environments and simple interface can drastically reduce the turnaround time from conceptual stage to functional system. However, *Gazebo* has some limitations, for example, physics models of soil, sand, grass, and other pliable surfaces normally found in nature are well beyond the scope of this project.

An example of the advantages of *Gazebo* was presented by Swanson *et al.* (2013). In this study, details of the design of a highly modular Hardware-in-the-Loop (*HIL*) driving simulator are proposed using the tools of open source software, such as *ROS* and *Gazebo*. The mention of a *HIL* driving simulator by the authors refers to a system where both a driver and a vehicle hardware component are interacting with each other. Because both software packages are fundamentally designed as *middleware*, *ROS* and *Gazebo* provide the ability to create unique features in the driving simulator. Swanson *et al.* highlights the effect of CPU choices, decreasing latency in the simulator and increasing system fidelity.

In (ECHEVERRIA *et al.*, 2011), the authors describe *MORSE* as an open-source application that can be used in different contexts for testing and verification of multi robotic systems as a whole, at a medium to high level of abstraction. *MORSE* is already used in real projects, demonstrating that it can fulfill the requirements specified of modularity, middleware independence and realistic visualization and physics. However, the simulation of multiple robots in *MORSE* can be limited by the processing power of the computer running the software.

2.1.2 Simulated systems

The prediction of the future path of ego-vehicle and other vehicles in the road environment is important for safety applications, especially for collision avoidance systems. Based on these scenario, in (LYTRIVIS; THOMAIDIS; AMDITIS, 2008) is proposed a cooperative path prediction algorithm that gathers position, velocity and yaw rate measurements from all vehicles in order to calculate the future paths. The core of the application is that each vehicle broadcasts periodically short messages so the ego driver can be aware of the current situation, specifically for the path that other vehicles are going to follow in the future. Based on this information, and on the on-board calculation of his future trajectory, the driver could be informed of imminent dangerous situations.

The algorithm that concerns path prediction is the same for both the ego-vehicle and the other vehicles. It is consisting of two main parts: *i*) it is extracted from the dynamic state

of the vehicle; and *ii*) it derives from the road geometry. In Lytrivis, Thomaidis and Amditis' study simulated data was used, and simulated scenarios were carried out for two main reasons: *i*) it is very difficult to have many equipped vehicles with inertial sensors, GPS, and wireless communication devices; and *ii*) safety limitations.

Doshi and Trivedi (2010) propose using a Layered Architecture for Cooperative Active Safety Applications (LACASA) that includes driver behavior (i.e., or intent prediction) as a fundamental building block for cooperative assistance systems. The scope of this research aims directly at the problem of understanding driver behavior and the drivers' interactions with the surrounding environment that influence their behaviors, in order to improve cooperative active safety systems. A Monte Carlo Simulation was performed to quantify statistics about how different levels of LACASA Driver Assistance Systems would affect the collision rate in such circumstances. The authors consider that the future of Intelligent Transportation Systems is inter-twined with the development and incremental implementation of distributed sensing and, communications networks. Therefore, a number of situations would benefit from intelligent cooperative advanced driver assistance systems, including Forward Collision Warning, Lane Change Assist. They conclude that *the great promise of vehicle to vehicle communications includes a reduction, or even elimination of collisions and fatalities on roadways, especially of those caused by driver error.*

Advanced Driver Assistance Systems (ADAS) based on communication between the Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) or Vehicle to Overall (V2X) have the potential to increase road safety and traffic efficiency. Nevertheless, these systems have to be analyzed carefully in order to optimize usability and avoid negative impact on drivers' workload, distraction and emotional response. For this reason, Maag *et al.* (2012) propose an analysis of a multi-driver simulation for two cooperative ADAS that assist the driver in merging assistant and hazard warning. By doing this, multi-driver simulation helps to combine various advantages of the traditional research methods: *i*) technical systems are not actually implemented but can be emulated; *ii*) the test subjects drive realistically; and *iii*) assumptions based on driving models are not needed. The instrument of multi-driver simulation is also very valuable for analyzing the effects of a hazard warning system on the behavior and safety of a group of drivers.

Experimental runs in real traffic allow studying authentic driving behavior of several road users in naturalistic situations. However, the uncontrollable conditions make the realization of test situations in real world traffic very difficult, and driving in safety-critical situations is highly problematic because of ethical reasons. With driving simulation it is possible to determine the effect of different traffic conditions on a human driver (MAAG *et al.*, 2012). This method is beneficial because conditions can be controlled, and desired test situations can be generated. Furthermore, due to the simulated conditions, the realization of safety-critical situations is possible. In Maag's study, the authors highlight that simulators still have a lot to improve when it comes to represent reality.

Laquai *et al.* (2013) introduce a way to extend the Intelligent Driver Model (*IDM*) to support multiple lanes and model driver imperfections. They used a system to support anticipative driving called Intelligent Support for Prospective Action (*ISPA*). The *ISPA* system aims to provide information about upcoming traffic situations that the driver potentially cannot anticipate. *ISPA* calculates three distances to an obstacle that are required to reach the target speed for the following deceleration strategies: *i*) stepping off the gas pedal (Coasting); *ii*) comfort deceleration by applying the brake (-0.2 g); and *iii*) strong deceleration (-0.4 g). Moreover, these study also describes the overall process of generating a multi-lane Car Following Model (*CFM*), based on the single lane Enhanced Intelligent Driver Model (*EIDM*). In order to run the required experiments three main tools were used: *i*) a driving simulator, *ii*) the *ISPA* system and *iii*) the traffic simulator SUMO. The main disadvantage of that study was the cost of the driving simulator tool.

Chen, Zhang and Ge (2013), consider a formation control problem under the constant spacing policy for the multi-vehicle systems with three fixed communication topologies. The authors describe different types of approaches to coordinate the strategy of control among vehicles, and they determine that the leader-following approach is the easiest to understand and implement. The formation control problem is to find a control input for each of the followers vehicles based on their available information such that the platoon of vehicles will travel in a steady formation. This study consider the use of a simulator, without a realistic ambient, to test the formation control of multi-vehicles and the effect of communication time-delay in the system proposed.

Guillet *et al.* (2014) present a general framework dedicated to adaptable robot formation control. It accounts for various kinds of dynamics encountered in a formation of several mobile robots (potentially heterogeneous), through an adaptive and predictive algorithm. The control architecture proposed in this study constitutes a generic strategy for the formation control of mobile robots for various applications in different contexts. To validate the multi-robot coordination and the mixed control, some simulations were realized, but the communication was not taken into consideration. In this study, on off-road situations, both terrain geometry and contact conditions may lead the robots to hazardous situations that might result in either a fatal roll over crash or a loss of controllability if the values delivered by the control laws exceed the actuators limits.

Control procedures and exchanged information for Cooperative Adaptive Cruise Control (*CACC*) was showed by Omae *et al.* (2014). In this study, they introduced a scheme for implementing *CACC* of heavy-duty vehicles developed as part of the NEDO's Development of Energy-Saving ITS Technologies project. This approach differs from previous *CACC* systems because it focuses on control procedures and contents of *CACC* inter-vehicle communication using broadcast communication. The vehicle and road models were developed in TruckSim, while the control procedures and inter-vehicle distance controllers were developed using Simulink.

Radars and inter-vehicle communication are also modeled using Simulink. The simulation results showed that the vehicles can form a platoon, exit the platoon, and permit vehicles cutting into the platoon from the adjacent lanes.

[Menhour et al. \(2017\)](#) describes the evolution of their developments on longitudinal and lateral vehicle control problem. An important part of this work is dedicated to present simulation results. The authors consider that the actual condition tests are often expensive and require specially equipped vehicles in order to study the interactions between embedded applications and events encountered in the road environment. The simulation allows an approach to the reality of a situation as it also allows the control of specific events and their reproducibility. In addition, the authors emphasize that the difficult or dangerous tests are easily accessible by simulation as well as the analysis of the functions' performance in a singular event, whereas in real condition, waiting hours could be required before the generation of such an event.

2.1.2.1 Vehicular communications

Vehicular communications are considered nowadays to be a paramount point of research for the development of technologies in the public safety of roads. Anti-collision warning, accident alarming and cooperative systems of assistance and control for drivers are some of the applications that this technology can offer to decrease the fatality rates of car accidents in the world. There have been several studies that have shown the beneficial use of simulation of wireless communication for vehicles in driving systems. Some of this previous works are described below.

[Gomes, Olaverri-Monreal and Ferreira \(2012\)](#) propose the See-Through System (*STS*), which is a cooperative system able to increase the visibility of drivers intending to overtake the preceding vehicle, making such critical maneuvers safer. By the authors, it is clear that a lane-keeping system is an efficient approach to mitigate head-on crashes caused by the inadvertent actions of drivers. The overtaking vehicle uses distance sensors and computer vision to determine the geometry of a partially translucent image that is displayed on its windshield, using laser holographic projection and overlapping the vision-obstructing vehicle. In this study, the authors developed a complex driver-centric simulator for the evaluation of *STS* based on vehicular ad-hoc networking (*VANET*), by coupling a realistic driving simulator to a *VANET* simulator, which tightly interconnects a microscopic traffic simulation engine with the *NS-3* network simulator. To function, *STS* requires that only the two vehicles involved in the overtaking maneuver are equipped with the system.

A decision-making module for cooperative collision warning was shown in ([ZARDOSHT; BEAUCHEMIN; BAUER, 2013](#)). In the study, this module was introduced for accident situations that process information from *VANET* communication and advises the driver an alternative route in order to avoid the traffic caused by the accident. The authors consider that, by obtaining information about the environment surrounding a vehicle, it is possible to assist a driver in

driving safer. This decision-making module has been implemented and tested using the Vehicles in Network Simulation (*Veins*). The authors consider that each car is equipped with *GPS* and wireless communication hardware to implement the decision-making algorithm. The information about the environment was shown using only one word (i.e., accident or released) plus the location of the vehicle involved. A disadvantage of such model is the possibility of overloading the driver with too much information.

In (JEMAA; GRUYER; GLASER, 2016), the authors propose an advanced simulation platform, which its main objective is to evaluate and validate cooperative perception applications in close-to-reality road scenarios. It interconnects several simulation tools in a single simulation framework. In this study, the authors consider that the anticipation of a dangerous driving situation requires extending the perception view of vehicles, and with the use of V2V communications this could be possible. Such platform addresses the need for a specific microscopic simulation view of the vehicular network for cooperative perception applications. It provides the possibility to test ADAS applications with accurate sensors' data running over the network simulator, *NS-3*. The platform uses real time sensor data provided by *Pro-Sivic* and processed in *RTMaps* for cooperative vehicular perception application.

2.1.3 Non-simulated systems

Urban driving environments present a variety of critical situations for drivers, and can result in collisions, injuries and deaths. In order to guarantee the road safety for people, some proposals are briefly described below.

Stiller, Farber and Kammel (2007) consider that the combination of a probabilistic reasoning framework with a formal logic language enables a cognitive automobile to handle uncertainties in measurements and contradictory rules. The authors propose to use ontological¹ concepts for a detailed description of traffic scenes, in order to secure that the complex knowledge base stays comprehensible and maintainable. Finally, the authors argue that in the long term automated cooperation among traffic participants bears the potential to improve traffic efficiency and safety beyond the level attainable by human drivers.

McCall *et al.* (2007) explore a vision system that estimates driver intentions in the specific act of lane changes, which is one of the most relevant actions to intelligent support systems. Moreover, they demonstrated a Driver Intent Inference System (DIIS) that is based on lane positional information, vehicle parameters and driver head motion. The *DIIS* is composed by a few key components: the lane position tracking system, the driver head motion estimation module, the vehicle parameter collection system and the lane changing intent classifier. These algorithms are then applied and evaluated on real-world data that are collected in a modular intelligent vehicle test bed. In McCall's study, the analysis of the data for lane changing intent is performed using a sparse Bayesian learning methodology. Unfortunately, there are many

¹ Ontologies are an explicit, formal representation of knowledge about a domain of application

problems with this approach, such as deciding when a true lane changing event occurs (i.e., when it begins or end).

According to [Papp, Brown and Bartels \(2008\)](#), cooperative intelligent vehicle systems constitute a promising way to improve traffic throughput, safety and comfort. This study describes the motivation behind the Local Dynamic Map (*LDM*) based on system decomposition approach, introduces the basic concepts of the world modeling dedicated to cooperative vehicle systems, and summarizes the main features of a prototype implementation. The consequences of the *LDM* based system architecture on application development and testing are also presented. The world model in the *LDM* is created based on high resolution digital maps, sensory inputs and communication. Moreover, the *LDM* implements an abstraction layer between the data interpretation and the higher level behavioral functions. This proposal was developed for the *SAFESPOT* Integrated Project. Finally, *the authors point out that testing and validation of cooperative vehicle systems pose significant challenges and safety risks*. Therefore, the architecture proposed supports high fidelity virtual testing via establishing links between the *LDM* and the simulated virtual world.

[Jerhot et al. \(2009\)](#) introduce a consistent modular approach to compute the optimal environment model and his confidence level, based on the available sensor measurements. This approach incorporates sensor data fusion with self-diagnosis capability, and estimates the maneuver level for the objects detected. The data fusion layer gathers all this information and fuses them into an *Environment Model*, which serves as an image of the current *real* environment. On the other hand, the diagnosis proposed for objects in the environment is based on tracked data of individual sensors. Finally, the intent estimation module computes the probability of defined maneuvers for the traffic participants. The Bayesian networks (*BN*) and thus the probabilities distributions, serve as a uniform and consistent denominator. As conclusion, the design of the *BNs* represents the fundamental challenge of the Jerhot's approach.

Considering [Schubert \(2012\)](#), complexity of Advanced Driver-Assistance Systems (*ADAS*) is steadily increasing. While the first applications were based just on warnings, the current systems actively intervene in the driving process. Due to this, the author considers that the application of Decision Networks (*DNs*) in the *ADAS* domain appears natural, because they facilitate a unified Bayesian view on the data-processing chain from the sensor to the decision stage. The algorithms proposed are illustrated on the example of a system that is capable of assessing the current traffic situation on highways to give lateral maneuver recommendations to the driver. In this Lane Change Assistant, the main requirement is that, regardless of the chosen maneuver, the host vehicle remains safe. The author concludes that one limitation for *DNs* is the disability to efficiently handle arbitrary continuous distributions. Moreover, this study considers that the validation of a probabilistic decision system may be a difficult task.

[Westhofen et al. \(2012\)](#) present an active vehicle safety system, based on transponder and camera data [i.e., Ko-TAG 1.0 and CMOS camera (752x480 pixels, 60fps, field of view

(FOV): 50 degrees)]. The system comprises tracking of pedestrians and vehicles, fusion of transponder and camera data, prediction of collision risk, warning of the driver, and autonomous breaking. Moreover, an experimental study of real-world scenarios demonstrated that the fusion of the sensor data results in a reliable prediction of upcoming collision risks and it enables a specific warning (i.e., or a justified autonomous brake maneuver) in order to avoid a collision. The collision risk with other traffic partners, especially pedestrians, it is assessed by using sophisticated motion models based on empirical data. Moreover, the collision risk calculation among vehicles and the driving tubes are determined by means of the linear bicycle model, considering physical constraints.

The behavior of cooperative systems is inherently more complex than the behavior of individual systems, due to the interactions of cooperative applications. It was one of the main reasons to create *test site*. Thus, [Passchier et al. \(2013\)](#) present the architecture of the road side facilities, and its compliance to standards, of the Dutch Integrated Test Site for Cooperative Mobility (*DITCM*). The *DITCM* test site has been developed over recent years as an extensive test site for cooperative systems in Europe in cooperation with many partners and projects. Moreover, *DITCM* operates an extensive set of facilities, including simulation environments, hardware in the loop facilities, and car labs. The fast density of sensors, communication equipment and application platforms, the flexibility to adapt to specific project requirements, and the possibility to use the same test site for open track and closed track tests, make the *DITCM* test site more suitable for rapid and flexible technical validation testing. However, the high cost of the infrastructure is its major disadvantage.

In the study presented by [Sivaraman and Trivedi \(2013\)](#) is discussed a cooperative and predictive driver assistance focused on cooperation in four different levels: in-vehicle systems cooperation, vehicle-driver cooperation, cooperation across vehicles (V2V), and cooperation with vehicles and infrastructure (V2I). The authors consider that cooperative active safety across vehicles, using V2V and V2I communication channels will quantitatively improve the performance of active safety systems. Moreover, *the transmitted dynamics will serve as a substitute for vehicle detection and tracking*, with active safety systems taking collision-avoidance action as necessary. Finally, authors explain how in the coming years, at the high end of the market, vehicles will be extensively instrumented for sensing, computation, and communication; whilst at the lower end, new vehicles will be equipped with wireless communication for V2V and V2I.

Examples of difficult urban driving scenarios include intersections, roundabouts, and merges. Due to this, [Sivaraman et al. \(2013\)](#) present a novel study focused on assisting the driver during merge maneuvers. In this study, given a robust understanding of the ego-vehicle's dynamics, the other vehicles, and the on-road environment, the system recommends merge maneuvers to the driver, formulating the maneuver as a dynamic programming problem over the Dynamic Probabilistic Drivability Map (*DPDM*), searching the minimum cost solution

for merging. In this system, the probability of drivability is determined from observation of the tracking information from the environmental sensors: camera, radars and LIDARs (Light Detection And Ranging). The camera is used solely for lane marker detection and lane tracking. The radars are able to detect and track vehicles as they overtake the ego-vehicle on either side. Finally, the LIDARs are able to estimate parameters such as vehicle length, width, and orientation, as well as position and velocity. The radars and LIDARs costs do not contribute with the economic system viability.

Appropriate human behavior is still a significant factor in traffic safety, at least until fully automated driving is achieved. Considering this, [Utsumi et al. \(2013\)](#) describe an ongoing project that aims to develop a collaborative safety mechanism based on mutual sharing of a driver's information (i.e., attention and performance). The authors believe that information would be very helpful for people to *predict/understand* another person's behavior in a traffic environment. To achieve this goal, they are working on three research topics: *i)* Driver's Behavior Evaluation; *ii)* Sharing Mechanism of Driver's Information; and *iii)* Assessment of Social Acceptance. Utsumi's study evaluates the driver's current performance at comparing driving behavior to predefined rules, which are determined based on interviews with professional driving instructors.

Moreover, Utsumi's study employs a behavior-based driver-evaluation method using multiple sensory devices for information detection. In addition, for information sharing, the authors are investigating both a method using vehicle-mounted displays and a method using wireless communication. On the other hand, the question about the proposed methodology is whether sharing information on a driver's reliability is acceptable to people. To answer this question, authors conducted a large-scale questionnaire investigation involving over 1,500 subjects (1,030 persons holding a driver's license and 515 persons not holding a driver's license). Furthermore, sharing "driver's attention" is considered especially effective in a situation where the observing drivers and pedestrians have already focused on a particular vehicle.

Finally, it was possible to find a reference architecture in the literature review. This architecture is independent of particular implementation technologies, tools and frameworks. It is an abstract set of solution considerations, requirements, patterns and guidelines, which are then concretized during an individual instantiation. In this context [Behere, Törngren and Chen \(2013\)](#) provide an engineering basis to develop cooperative driving systems. The authors argue that the research on cooperative driving area can be broadly split into two categories: *i)* research within the areas of automatic control, wireless communication, and smart transport infrastructures (i.e., this research aims to develop the key knowledge and technologies needed to make cooperative driving possible), and *ii)* research needed to integrate the individual technologies and develop comprehensive cooperative driving solutions. The authors of this study also consider that information sharing between vehicles and information sharing between vehicles and the infrastructure are the most attractive trend in intelligent vehicle research.

2.2 Final considerations

Important characteristics of the studies analyzed in this chapter can be summarized as follows:

- The fidelity of the simulation platforms, for experiments with multiple vehicles, can be affected by the characteristics of the hardware used in the execution of the simulations. Another important point to highlight, with respect to the simulation platforms, is the fact that, these platforms still have the challenge of representing, in the best possible way, the physical phenomena of an environment;
- Most of these studies have been developed using simulation tools, due to its safety validating the cooperative system performance using these tools. In the economic aspect, some simulation platforms may not be adequate, due to the costs that can reach these tools. However, there are currently some open-source simulation platforms, with which it is possible to develop new systems;
- With the use of simulation platforms, a cooperative system could be emulated. However, the physical limitations that can be found in real situations must be considered;
- Input information that are mostly considered by cooperative systems are the self-information (i.e., ego-vehicle data) and the shared information from nearby vehicles;
- Evidences obtained from the analyzed studies in this chapter allow for two conclusions: *i)* that to develop cooperative systems it is not necessary to use sensors like radars and lasers. Instead, those systems can be developed using other sensors, such as cameras, GPS, and inertial sensors; and *ii)* the advantages that the use of wireless communications can represent (i.e., V2V or V2I); and
- Finally, in the studies considered in this chapter, it is possible to highlight the use of stochastic methodologies, for the development of cooperative driving assistance systems.

In this chapter, it was considered the combination of each of the characteristics mentioned above, in order to solve the given problem in [section 1.1](#). With the integration of this information our approach has been able to complement some of the limitations of the cited studies (i.e., economics, safety, validation and modeling), thus highlights the contribution of our approach to the area. In chapters [3](#), [4](#) and [5](#) we will have developed the ideas obtained from the studies cited in this chapter.

DRIVING SIMULATION PLATFORM

Considering [Schöner and Morys \(2016\)](#), different concepts for driving simulation platform have been realized. However, all of them depend on the application. Schöner et al. emphasizes that currently, there are numerous technical realizations of driving simulators, from a static flat screen with a simple steering wheel and pedals to large dynamic simulators with perfected immersion technologies. Within some applications for the driving simulators, it is possible to find:

- Functional vehicle demonstrations;
- Investigations in control concepts;
- Training for drivers;
- Accident research;
- Study of driver performance and development of driver models; and
- Testing and verification of driver assistance systems.

In addition to the considerable cost and high level of logistics, ensuring the safety of participants is another key factor in conducting real tests. Considering the complexity involved to perform this type of test, and based on the studies analyzed in the previous chapter, specifically the proposals of the subsection [2.1.2](#), we verified the need to use a Driving Simulation Platform for this thesis.

Therefore, to meet the purposes and perform the simulations of this thesis, it was important that the platform had three important characteristics: *i*) to create; *ii*) to control and *iii*) to reproduce road events that may happen in real situations. In addition, the platform should be open-source (i.e., for economic reasons), with the flexibility of allowing it to be adapted when it is needed. Taking into account that the Mobile Robotics Laboratory (LRM) already had a work

on an initial stage under a similar assumption of our thesis, we decided to extend that work so that it would correspond to the requirements of the new proposal elaborated by us. The current version of this platform is an extension significantly different model from the initial version ¹.

In this chapter, we present in detail the driving simulation platform used. With the development of the architecture proposed for this platform, users have the opportunity to perform simulations that are contained in the list of applications suggested by [Schöner and Morys \(2016\)](#). Examples of the application and evolution of this platform can be found in chapters 4, 5 and 6 of this thesis.

3.1 History

The idea of using simulators in driving training issues originates from the first flight simulator which was developed in the early 1910s ([HASSAN, 2014](#)). This first simulator was created to be used for training pilots and for reducing operating costs required, compared to the use of real equipment ([BLANA, 1996](#)). The widespread use of flight simulators inspired researchers to apply the same concept for road vehicles ([HASSAN, 2014](#)).

Highway research simulations were developed and operated in the late 1950's and in the early 1960's. However, the technological challenges that were present at that time (e.g. visual displays and computer technology) prevented the advance of the highway simulators in the middle of the 60's. In the following years, much of the technology needed was developed by the National Aeronautics and Space Administration (NASA) to support its space program ([ROBERTS, 1980](#)). In the 1970s, there were at least 20 research driving simulators throughout the US and Europe, including many small devices used for drivers' training and licensing ([FISHER *et al.*, 2011](#)).

In the successive decades, the driving simulators were rapidly evolving due to the development of digital computers (e.g. more powerful Central Processing Units "CPU" and Graphical Processing Units "GPU"). This advance in technology allowed the development of more complex models with 3D simulations represented in real-time at a low-cost.

3.2 Classification of driving simulators

The significant difference among driving simulators is focused essentially in the characteristics of the components used in their fabrication. These components define the level of fidelity in a driving simulator, and, for this reason, there is a direct economic dependence. *Fidelity* describes the degree to which the simulator's characteristics faithfully replicate the driving task ([FISHER *et al.*, 2011](#)). Therefore, the driving simulators can be classified as low, medium, and

¹ The basic idea of the driving simulation platform was presented by the master student Tiago Cesar Dos Santos in 2013.

high according to their acquisition cost. Sometimes they can also be referred to as low, mid, and high level driving simulators (LANG *et al.*, 2007).

3.2.1 Low-cost driving simulators

They are called low-cost due to the relatively inexpensive graphic displays. Typically, they have a single display which provides a narrow horizontal field of view and a gaming steering wheel as a Human Machine Interface (HMI). These driving simulators do not have a motion system and their fidelity is restricted (BLANA, 1996; HASSAN, 2014).

3.2.2 Medium-cost driving simulators

These driving simulators employ advanced imaging techniques using real-time animation to create a scene that is projected in front of the driver, a large projection screen (or a set of screens), and a modified car (or a car) with all or some of the regular controls (BLANA, 1996).

3.2.3 High-cost driving simulators

These driving simulators provide a wide field of view (i.e., in some cases a 360-degrees view) and an extensive motion system. They can include more than six degrees of freedom and the capacity of loading a real car or parts of it; thus, having the best fidelity of them all. Examples of high-cost driving simulators are the Daimler-Benz and the VTI driving simulators in Europe and the National Advanced Driving Simulator (NADS) in the United States.

The sound is an aspect found in all three simulators. The differences are observed in some details like location, sound quality, and the capacity to create the sounds of events in the driving simulation scenery (e.g. wind, cars, brake, and so on).

It is important to consider that the relation of cost-benefit in the driving simulators depends on the goal and on the level of details needed to represent the real world. In Figure 4 Smith *et al.* emphasized that the best *test track* in a simulation has to combine the realism with the controlled situations, in function of the acceptance and safety (SMITH; WITT; BAKOWSKI, 2008). For this reason, it is common to find driving simulators used in researches with a representative cost, regarding the use of a simulator for training.

3.3 Advantages and disadvantages of driving simulators

A major disadvantage of a driving simulator is that it isn't able to represent the complexity of real road situations. For this reason, the validity and reliability of these tools are often questioned in the research area. However, despite of this disadvantage, automotive industry (AEBERHARD *et al.*, 2015; ZEEB, 2010; CHALLEN, 2008) and research centers (RUDIN-BROWN; WILLIAMSON; LENNE, 2009; STAVRINOS *et al.*, 2013; KNAPP *et al.*, 2009;

Figure 4 – Selection of the best *test track* for an application, in function of the realism and the control.

Source: Smith, Witt and Bakowski (2008).

GIETELINK *et al.*, 2006; GONCALVES *et al.*, 2014; MAAG *et al.*, 2012; BIFULCO *et al.*, 2012; NOTH; EDELBRUNNER; IOSSIFIDIS, 2012; YOSHIMOTO; SUETOMI, 2008) have a driving simulator available to facilitate the design, development, and test of their proposals.

3.4 Validation of driving simulators

The models used in any driving simulator have to pass through tests and validations regarding the system, entity or idea that is being represented. The objective is to obtain a level of confidence in the driving simulator (ESPIÉ; GAURIAT; DURAZ, 2005). Engen (2008) considers that the driving simulators have to guarantee that the results obtained can be replicated, at least in a high percentage, in a real field test. However, it is not always possible to compare the results of a driving simulator with the results of a real field of test (generally, for economic and logistic limitations), therefore is needed to find other alternatives. Engen proposes to consult a larger array of sources in order to find such alternatives and evidence of validity to be included in research models.

3.5 Driving simulators used by the automotive industry

The automotive industry use driving simulators to test a new system on a vehicle and to analyze how the system could influence the driver. For this reason, the driving simulation platforms used by the automotive industry have the highest possible fidelity. Some examples of driving simulators used by the automotive industry are the ULTIMATE by Renault's, Dynamic Driving Simulator by BMW, the Virtex by Ford, the Daimler AG driving simulator by Mercedes, among others.

3.6 Driving simulators used in researches

In researches, the driving simulators have focused on the analysis of drivers' behavior regarding some hazard situations, considering specific factors that could affect the driver (e.g. mobile phones, psychoactive substance use, test traffic signals, test driver assistance systems, among others). Considering [Castro \(2008\)](#), some variables measured by the driving simulators in researches are: *i)* vehicle speed; *ii)* vehicle place on road markings; *iii)* distance from the vehicle in front; *iv)* angle of the steering wheel; and *v)* amount of pressure applied to the brake pedal. Therefore, the driving simulators used in researches have a high fidelity. In researches the nature of the experiments is always dynamics (i.e., it may be necessary to change some characteristics for the driving simulation platform), and for this reason, the cost and performance of the awaited results could be compromised. Therefore, when a driving simulator is used in researches, it is necessary that the system structure be re-configurable ([HASSAN, 2014](#)).

Nowadays, there are multiple driving simulation platforms to develop and evaluate safety and emergency systems. Some of them are commercial and are more prominent by its technical characteristics, such as: *i)* the exact dynamic model used in the vehicles; *ii)* integration with other software tools; and *iii)* the fidelity of obtained results. Moreover, some of these platforms are used by some automotive industries and research centers ^{2,3}. However, these driving simulation platforms have considerable acquisition and maintenance costs. Furthermore, the commercial driving simulation platforms also have a closed code, which allows only a minimal tool adaptation. Among some commercial driving simulation platforms most known are: *CarSim* and *TruckSim* from *Mechanical Simulation*, *PreScan* from *TASS International*, and *CarMaker* from *IPG Automotive GmbH*.

There are also state of the art open-sourced driving simulation platforms, which, in addition to the economic benefit, allow the simulation platform to adjust to the needs of the user. However, one of the biggest drawbacks of these simulation platforms is that they do not provide users with adequate technical support. For this reason, they require a longer learning time. Among some open-source driving simulation platforms there are: the *TORSC* ([WYMANN et al., 2014](#)), which is a highly portable multi-platform car racing simulation and used as an ordinary car racing game, as an *AI* racing game and as a research platform; and the *Delta3D* ⁴, which is a game and simulation engine appropriate for a wide variety of simulation and entertainment applications.

The driving simulation platform presented in this thesis is an open-sourced tool, with a middle level of fidelity (See [section 3.7](#)). Its objective is to contribute to the development of new safety and emergency systems in the first stage of modeling. Similar to other open-sourced driving simulation platforms, this tool allows the users to configure its architecture by using

² <<https://www.carsim.com/products/ds/index.php>>

³ <<https://www.tassinternational.com/testimonials>>

⁴ <<http://delta3dengine.org/>>

different software tools (i.e., also open-source) that integrate it. The integration of the driving simulation platform with different software tools could be considered its main advantage.

3.7 Driving simulation platform architecture

The previous section had as objective to consider some relevant aspects of the driving simulators and their applications. For example, we assumed that the relation between the realism and the control of the simulated situations in this driving simulation platform could be located in the middle of the *testtrack's* circle, as shown in Figure 4. The main reason for why this particular point was selected in the *test track* is due to the lack of a adequate level of realism in the models used in the current platform version (i.e., exact vehicle dynamics and environmental conditions of the simulation scenario, among others). Taking into account this lack of realism, it is recommended the use of this simulation platform only in the first stage of the system development. The obtained result using the presented driving simulation platform could help the platform users to consider a test of one's system in real field-tests.

The driving simulation platform proposed is divided in three main parts: *i*) a generic simulator for academic robotics; *ii*) a middleware for robotic applications; and *iii*) a communications simulator. The generic simulator for academic robotics used is the Modular Open Robots Simulation Engine (MORSE) (i.e., the version used in this thesis was morse 1.2.2). *MORSE* is the main tool for the driving simulation platform. As described by Echeverria *et al.* (2011), *MORSE* is a free and open-sourced software tool, which can be adapted to the requirements of a particular system. *MORSE* has a large number of sensors that can be used in the development and evaluation of the safety and emergency systems. Furthermore, *MORSE* proposes not only some methods to alter the input and output data (i.e., through modifiers) but also allows the creation of new sensors⁵, such as the Network Interface Communication (*NIC*) sensor, which was used in (GÓMEZ *et al.*, 2014; GÓMEZ *et al.*, 2016). The *NIC* sensor has the function to simulate a communication vehicular card.

To perform simulations on the driving simulation platform, we use one of the fundamental features of *MORSE*, which consist in creating small python scripts. In these scripts, each of the components (i.e., vehicles, sensors, and scenarios) required in the simulation is established. Whereas *MORSE* is an application developed entirely in python, much of its source code can be easily changed (ONERA; LAAS-CNRS; ISAE-SUPAERO, 2017). This feature facilitates the creation of some functions, which were used in the configuration of vehicles in the scenario, for example, the initial vehicle parameters (i.e., velocity, lane, and displacement path) can be directly configured in the simulation run scripts.

The simulations in *MORSE* are based in *Blender*, which is a tool with the main purpose of creating images and 3D computer animations through a high level of graphic detailing (i.e.,

⁵ <<https://www.openrobots.org/morse/doc/stable/contributing.html>>

the version used was blender-2.73a). This characteristic of *MORSE* allows the creation of different 3D vehicle models (Figure 5) with some dynamic characteristics (subsection 3.7.1). With *Blender* it is also possible to create virtual scenarios for the simulations. In this proposal, were considered realistic scenarios to create virtual scenarios, using GPS coordinates that were plotted into the *Blender* version used.

Figure 5 – The 3D simulation model of CaRINA 2.

Source: Santos *et al.* (2015).

In the driving simulation platform, it is also used the Robot Operating System (ROS) (QUIGLEY *et al.*, 2009), a middleware used in robotic that has, as a main feature, libraries and tools to help software developers create robot applications (i.e., the version used was indigo). Given that *MORSE* allows an easy integration with *ROS*, it is possible to get data of each one of the sensors in every vehicle during a simulation. Through *ROS*, it is possible to implement models (e.g., control, *DAS*, among others).

Finally, the communications simulator used in the proposed driving simulation platform architecture is *NS-3* (i.e., the version used was ns-3.19). With *NS-3* it is possible to develop codes using the programming languages *C++* or *python*. Besides being open-source, *NS-3* contains some tools for analysis of networks (e.g., *pcap*), and it is also possible to find different models for computer networks (e.g., propagation and Wi-Fi among others). For example, in this study it was used the model Wireless Access in Vehicular Environments (*WAVE*), to simulate the vehicle to vehicle communication *V2V* used in Chapter 4. In the simulation using *NS-3* the vehicular communication *V2V* has a node (i.e., in *NS-3* a node is a computer device that connects with a network) that represents a vehicle of the simulation executed in *MORSE*. Figure 6 presents one example of the architecture's configuration proposed in this thesis.

Figure 6 – Simulation platform architecture example.

Source: Gómez *et al.* (2014).

In Figure 6 is possible to observe that the three main parts of the driving simulation platform architecture communicate with each other by using two *sockets* (i.e., socket⁶ is a software endpoint that establishes bidirectional communication between a server program and one or more client programs). In the proposed driving simulation platform architecture *NS – 3* requests information to *MORSE* through the *Socket – SS*. When *MORSE* receives the request, *MORSE* responds to *NS – 3* with a determined kinematic package data of the vehicle that has requested it. The answer of *MORSE* is sent using the *Socket – SS*. Then, *NS – 3* receives the kinematic package data sent by *MORSE* and directs the information to the correspondent node. Finally, the information received in *NS – 3* is sent to *ROS* using the *Socket – MS*, with the objective to be used in a model. Data received by *NS – 3* are considered package in network simulation. This process is used by each node during all simulations.

3.7.1 Vehicular dynamic model

The actual Vehicular Dynamic Model (*VDM*) used in this driving simulation platform is based on the *Bullet physics* engine adopted for *Blender*. *Bullet Physics* is an open-sourced library, written in portable C++, used for professional collision detection and for rigid and soft body dynamics. This library is primarily designed for the use in games, visual effects, and robotic simulation (COUMANS, 2015).

Bullet's approach to a vehicle controller is called a *Raycast Vehicle*. A *Raycast Vehicle* works by casting a ray for each wheel. By using the ray's intersection point, we can calculate the suspension length and, consequently, the suspension force that is then applied to the chassis, keeping it from hitting the ground. The friction force is calculated for each wheel where the ray

⁶ The JavaTM Tutorials

contacts the ground. This is applied as a sideways and forwards force (BLENDER, 2016). A detailed example of the entire implementation of the *VDM* used in the vehicles of this driving simulation platform can be found at <http://www.tutorialsforblender3d.com/Game_Engine/Vehicle/Vehicle_2.html>.

3.8 Final considerations

In this chapter we have presented a brief description of the general context of the driving simulation platforms where the necessary requirements of these tools, as well as their main advantages and disadvantages, were analyzed. Taking into account this information, we present the architecture of the driving simulation platform developed, in which its ability to be reconfigured is highlighted, depending on the characteristics that may be required. Since its initial elaboration, the architecture of this platform had as main objective to allow the users the integration of free software tools. The realization of this objective can be verified in the integration made in the platform architecture with the communications simulator *NS – 3*. This integration of open-source tools allowed us to develop and evaluate each of the driver assistance systems that will be presented in this thesis.

COOPERATIVE INTELLIGENT TRANSPORT SYSTEMS

Considering [Horton et al. \(2016\)](#), [EU \(2016\)](#), the Cooperative Intelligent Transport Systems (C-ITS) allow to the street users (i.e., drivers, pedestrians, among others) and the land traffic managers to share information. This information is distributed through the use of V2V and V2I communication technologies. The main purpose of this exchange of information is to achieve better coordination of actions between them. An example of applying these systems is the warning of potentially dangerous situations among drivers. In this particular case, the C – ITS can help drivers in making decisions, and in the future could be done by the vehicle itself.

The continuous development of vehicular communications, as well as the automobile industry, is enabling the consolidation ([BHARATI; ZHUANG, 2017](#)) of approaches, such as the Vehicular Ad Hoc Network (VANET), which focus mainly on improving road safety. According to *NHTSA*, road traffic accidents, specifically of light vehicles, could decrease by approximately 80% with the use of road safety applications enabled through *VANETs* ([HARDING et al., 2014b](#)).

In the case of *VANETs*, [Lu et al. \(2014\)](#) and [Luan et al. \(2014\)](#) consider the following three groups: *i*) Safety applications; *ii*) Traffic management applications; and *iii*) Infotainment applications. Based on these groups, the main contribution of this thesis is the development of a safety application.

Although the application of *VANETs* on the roads is of great importance there are many challenges in its implementation. Some problems such as high vehicle mobility, variations in speeds, congestion on the roads and limited communication coverage between vehicles can cause constant changes in the dynamic topology of the network and, therefore, loss of connection ([BAI; STANCIL; KRISHNAN, 2010](#); [MECKLENBRAUKER et al., 2011](#); [KAREDAL et al., 2011](#); [ALASMARY; ZHUANG, 2010](#)).

In addition to these challenges, some of the applications allowed by *VANETs* require strict quality requirements (*QoS*) in the communication protocols. An example of this are the safety applications. In the safety applications, it is necessary to consider the delay times in the $100ms$ range, which is difficult to guarantee given the dynamic conditions of the network (GÓMEZ *et al.*, 2016). Considering these adversities presented, *VANETs* require an efficient, adequate and robust broadcast service to be able to use the safety applications successfully (BHARATI; ZHUANG, 2017).

To meet the objectives of this thesis, the *VANETs* were treated as a tool for the development of our proposal: creation of a safety application for motorists on the highways. With this in mind, solving the challenges present in *VANETs* is not part of the objectives of this thesis.

Finally, this chapter is a partial reprint of material presented in IEEE Intelligent Transportation Systems 2014 and 2016. These studies were made in collaborative work with LRM members¹, and the professors Arthur de Miranda Neto², Sebastien Glaser³ and Yasser Alayli⁴.

4.1 Simulation platform for cooperative vehicle systems

The variety of applications of the cooperative vehicle systems and their developments are also increasing rapidly (PASSCHIER *et al.*, 2013). For instance, the Cooperative Adaptive Cruise Control system (*CACC*) is an enhancement of the Adaptive Cruise Control system (*ACC*) that uses inter-vehicle communication to perform safe cruising at shorter inter-vehicle distance (OMAE *et al.*, 2014). In order to validate the proposed driving simulation platform architecture, a *CACC* was used since it requires both *V2V* communication and the simulation of multiple agents. A state-of-the-art controller based on (AREM; DRIEL; VISSER, 2006) was used.

4.1.1 Control

Given that $[\Delta x_i, \Delta v_i, \Delta a_i]^T$ are the distance, relative speed and relative acceleration between the i -th and the $(i - 1)$ -th vehicle of a platoon, respectively, Δx_i^{safe} is the safety distance between vehicles given by

$$\Delta x_i^{safe} = h v_i + d \quad (4.1)$$

where h is the headway time and d is the desired distance for a stopped vehicle. From

¹ Tiago C. dos Santos, Paulo V. Prado, Juan C. Perafan, Diego Gomes, Carlos Massera Filho

² The Terrestrial Mobility Laboratory, Federal University of Lavras, arthur.miranda@deg.ufla.br

³ The Institute VEDECOM, Sebastien.glaser@vedecom.fr

⁴ The Versailles Engineering Systems Laboratory, University of Versailles Saint-Quentin-en-Yvelines, yasser.alayli@uvsq.fr

the desired safe distance, it is possible to define the control error e_i :

$$\mathbf{e}_i = \Delta x_i - \Delta x_i^{safe} \quad (4.2)$$

and its derivative $\dot{e}_i$:

$$\dot{\mathbf{e}}_i = v_{i-1} - v_i - h a_i \quad (4.3)$$

Assuming that the acceleration is directly controllable and the jerk is unbounded, a CACC can be defined as a Proportional – Derivative (PD) controller:

$$\Delta a_i = K_p(\Delta x_i - h v_i - d) + K_d(\Delta v_i - h a_i) \quad (4.4)$$

Where, K_p and K_d are the coefficients for the proportional, and derivative terms, respectively. However, implementing it as a pure PD controller term would cause instability depending on the operation frequency since the current acceleration would be based on the previous one. In order to avoid this scenario Equation 4.4 can be rewritten as:

$$a_i = \frac{a_{i-1} + K_p(\Delta x_i - h v_i - d) + K_d(\Delta v_i)}{1 + K_d h} \quad (4.5)$$

4.1.2 Driving simulation platform configuration

Figure 7 – Driving simulation platform configuration used in the cooperative vehicle system.

Source: Elaborated by the author.

Figure 7 shows the configuration of the driving simulation platform used for the implementation and evaluation of the control's model present in the subsection 4.1.1. In the

configuration of its architecture, it is possible to observe the three main parts of the driving simulation platform, described in [section 3.7](#). In [Figure 7](#), the *Robotic Simulator* rectangle represents the *MORSE* simulator. In that rectangle, it is possible to observe the vehicles (robots Leader, Follower1, Follower2 and Follower3) used in the validation of the *CACC* control (see subsection [4.1.3.2](#)). In each vehicle simulated in *MORSE*, sensors (i.e., *GPS*, *velocity*, *pose*) were added to obtain data about velocity, position and acceleration of the vehicles. Each information was used in the control model.

In [Figure 7](#) it is also possible to observe that each vehicle in the *Robotic Simulator* uses a Network Interface Communication sensor *NIC*. The *NIC* sensor creates a message with different data [i.e., ID vehicle, position (x,y,z), yaw, velocity and acceleration] of each simulated vehicle in *MORSE*. This message is shared with the *Communication Network simulator* rectangle that represents the *NS – 3* simulator. The sending of messages between the *Robotic Simulator* and the *Communication Network simulator* is done through the use of the *SOCKET-SS*. The sockets' function shown in [Figure 7](#) is detailed in the last paragraph of the driving simulation platform architecture description, before subsection [3.7.1](#).

The communications model of vehicle to vehicle (V2V) implemented in the communications simulator *NS – 3*, consists in a precedent communication's topology proposed by [Chen, Zhang and Ge \(2013\)](#). The V2V communication is developed using four nodes, which are represented by each one of the vehicles used in *MORSE* (Leader, Follower1, Follower2, Follower3). Each node in *NS – 3* has an 802.11p Wi-Fi physical layer, a *MAC* layer and a definition of the hardware's physical properties. Moreover, for each node in *NS – 3* was assigned an *IP* address with the purpose of sending data packages between nodes. When *NS – 3* simulator obtains a message shared by the sensor *NIC* in *MORSE*, the V2V communications model sends the obtained message to the corresponding vehicular node. This process is repeated throughout the system validation *CACC*.

After a node receives a message through *NS – 3* it resends the message to the *MIDDLEWARE* rectangle that represents *ROS* using the *SOCKET-MS*. At this moment begins the execution of the control model presented in subsection [4.1.1](#). As it is described in the [section 3.7](#), *ROS* is a middleware that allows the implementation of models. It is divided into a set of nodes, and inside of this set there is one main node called *Master*. The *Master* node initializes *ROS* and allows the others nodes (i.e., these nodes are pieces of software) to communicate with each other. The *ROS* nodes are different from the nodes used in *NS – 3*.

In the validation of the control's model with the driving simulation platform seven nodes were implemented in *ROS*: *i*) one node to get the sent messages through the *SOCKET-MS* (represented by the *Network simulator Output* inside of the *Middleware* rectangle); and *ii*) a pair of *CACC controller* nodes ([Figure 7](#)) for each of the three *Follower* vehicles (with the purpose of determining the behavior of each *Follower* vehicle, based on the obtained messages from the *Network simulator Output* node). Finally, the control's model effect can be observed in each

Follower vehicle throughout the execution of the simulation in *MORSE*.

4.1.3 Results

As a validation of the driving simulation platform proposed in this study, two experiments were performed with the CACC control described in subsection 4.1.1. In every experiment we assumed that the vehicles had access to their own position and velocity information. A video with the execution of a test is available in the website <<https://www.youtube.com/watch?v=WQsRR9ajiPc>>.

4.1.3.1 Delay effect on the CACC system

In the first experiment, two vehicles were used. The follower vehicle must keep a safe distance from the leader vehicle using the longitudinal control law described in subsection 4.1.1. The purpose of this experiment was to evaluate the effect of the communication delays on the control of the follower vehicle. The scenario used for the experiments was an open space with no obstacles.

The tests have been performed using different wireless communication delays of 0.01s, 0.1s, 0.5s, and 1.0s. Figure 8 and Figure 9 show the velocity of both vehicles obtained from the network simulator and the histogram of the difference in the velocity of the two vehicles, directly from the output of the *NS* – 3. The total time of the experiment was 33s: From 0s to 10s, the leader vehicle accelerated from 0m/s to 37m/s; from 10s to 15s, the velocity of the leader decreased to 8m/s; from 15s to 25s, it increased again to 37m/s; and from 25s to 33s, the leader decelerated until it achieved 3m/s. Figure 8a and Figure 8b show the velocity profile and histogram, respectively, of the vehicle's velocity obtained from the network simulator with a 0.01s communication delay. We can notice that the follower's velocity follows very close to the leader's. Meanwhile, Figure 9a and Figure 9b present the velocity profile and histogram, respectively, from the simulation with a 1.0s communication delay. The higher communication delay results on a non-continuous velocity profile. Also the histogram presented in Figure 9b shows a larger variance in the velocity difference when compared to the data presented in Figure 8b.

Table 1 shows the average results of the velocity difference of both vehicles obtained from the network simulator with communication delays of 0.01s, 0.1s, 0.5s, and 1.0s. It is possible to notice that the maximum velocity variation increases with higher delay values.

Figure 8 – Leader vehicle Velocity (L , black), Follower vehicle velocity (F , red), velocities variation ($L-F$, blue), communication delay 0.01 s.

Source: Elaborated by the author.

Figure 9 – Leader vehicle Velocity (L , black), Follower vehicle velocity (F , red), velocities variation ($L-F$, blue), communication delay 1.0 s.

Source: Elaborated by the author.

Table 1 – Statistics features of L-F velocity variations due to delay times in communication.

Delay (s)	μ (m/s)	σ (m/s)	σ^2 (m/s) ²	Max. variation (m/s)
0.01	0.0630	1.4620	2.1374	3.276
0.1	0.0540	1.2684	1.6089	3.582
0.5	0.0263	1.1926	1.4224	6.091
1.0	-0.0121	1.7828	3.1786	8.007

Source: Research data.

4.1.3.2 CACC system validation

In the second experiment four vehicles were used (one leader L and three followers $F1$, $F2$ and $F3$) to evaluate the CACC performance. Each follower vehicle only communicated to the vehicle immediately ahead of it. Figure 10 and Figure 11 present the simulation results for communication delays of 0.01s and 0.333s. Figure 10a and Figure 11a present the velocity of each vehicle over time for the two delay settings. In Figure 10a, it is possible to notice that when the leader is accelerating its velocity is higher than the velocity of the followers. During the leader's deceleration we have the opposite behavior. It happens due to the small delay in the control response, which is caused by the inertia of the vehicle, communication delay and other physical aspects of the simulation. In Figure 11a, it is possible to observe how the behavior of the three followers is affected by the increment of the communication delay among vehicles. Consequently, an irregular behavior in each acceleration/deceleration pattern can be noticed.

Figure 10b and Figure 11b show the distance between the vehicles over time with communication delay of 0.01s and 0.333s, respectively. As stated in the CACC control law, the distance varies according to the velocity of the vehicles. Nevertheless, it is possible to see that the distances between vehicles $L-F1$, $F1-F2$ and $F2-F3$ have the same profile. When the communication delay of 0.333s was used, an increase in the difference of distance among vehicles of CACC system was observed.

Studies consisted of experiments done in a real field-test, with similar characteristics to the experiments done with the proposed driving simulation platform, were used as a reference in order to contrast the results related to the effect of the time delay in the CACC control. This decision is justified in subsection 3.4.

Among the different studies researched, the most valuable was the one by Ploeg *et al.* (2011), where the authors proposed a design and validation of a Cooperative Adaptive Cruise Control (CACC) in a real field-test. In its experiment a fleet of six Toyota Prius III were used since this model presents favorable technical characteristics for a real field-test. These vehicles were also equipped with other components in order to configure the CACC system appropriately [section IV of Ploeg *et al.* (2011)].

Figure 10 – Velocity and distance between the vehicles with communication delay 0.01 s.

Source: Elaborated by the author.

Figure 11 – Velocity and distance between the vehicles with communication delay 0.333 s.

Source: Elaborated by the author.

Ploeg et al. considered the inclusion of time delay when developing the control implemented in the field-test [section III-B of Ploeg *et al.* (2011)]. According to the authors, the time delay may produce disturbances in the control of vehicles (i.e., string stability), which can also be observed in Figure 11 of this study. For this reason, Ploeg *et al.* (2011) considered a time delay of 150ms in the communication system. The magnitude of such time delay was considered to be the best suited by the authors for their experiment. This value of time delay was also favorable to our results. Table 1 in subsection 4.1.3.1 of this thesis shows that a short time delay presents lower variations in the CACC system. The obtained results in the CACC validation proposed by Ploeg et al. can be found in section V-B of their study (PLOEG *et al.*, 2011).

4.2 Cooperative collision warning for driving assistance

This contribution is focused on proposing and evaluating a new alternative for driving assistance, based on the use of a Vehicle to Vehicle (V2V) communication. This system can serve to warn and prevent collisions among vehicles. The proposal contribution consists in the mathematical reasoning developed by us, which determines vehicles' presence around the ego vehicle. In this contribution, a new version of the simulation framework proposed in (GÓMEZ *et al.*, 2014) also is presented. This new version allows vehicular communication to be closer to reality, by means of bidirectional exchange of messages among vehicles and a time delay in dependence of the distance among them.

4.2.1 Model

Collision Warning (CW) for driving assistance has been the focus of studies (CHO *et al.*, 2009; ZHU; SINGH, 2010; SEBASTIAN *et al.*, 2009) in the last decade. Generally, the CW models are divided in three main parts: *i*) perception of vehicle's environment; *ii*) collision compute; and *iii*) presentation of the collision compute results for the driver. In relation to the *perception of vehicle's environment*, generally some data from the surrounding vehicles (e.g., position, velocity, heading, among others) may be obtained. In this stage, some sensors (i.e., laser, radar, and so on) are used to get information from the closest vehicles. However, when it comes to the use of sensors, the coverage range and its performance in each one of the different work environments may be an inconvenient for the driver assistance system operation regarding the cost.

In relation to the *collision compute*, some kinematic approaches that allow taking the information obtained from *perception of the vehicle's environment* are used normally. In some studies (CHO *et al.*, 2009; ZHU; SINGH, 2010; SEBASTIAN *et al.*, 2009; XIANG; QIN; XIANG, 2014; HUANG; LIN, 2013; ZARDOSHT; BEAUCHEMIN; BAUER, 2013), a set of assumptions (e.g., sensors type, sensors data output, or map use, among others) that allow to describe the model are also considered. Finally, the obtained results in the *collision compute* have

to be *presented for the driver*, with the objective of notifying the driver about the possibility of a risk situation in the vehicle's environment. Some examples are: *i*) screens where are presented warnings or sounds indicating danger are presented; and *ii*) devices that vibrate as a warning for danger. Some safety benefits from the *CW* models application can be appreciated in (KUSANO; GABLER, 2012), where it is analyzed the effectiveness of the benefits using three pre-collision systems.

Since the last decade, the use of vehicular communications on safety applications has been considered an alternative to complement the Driver Assistance Systems. This can be observed in (BARBARESSO *et al.*, 2014), where it is highlighted the *USA* commitment to implementing this type of technology in the country, given the significant results obtained. Two strong reasons for this growing interest can be named: *i*) the fact that vehicular communication allows a 360-degrees coverage of the vehicle; and *ii*) the perception of long distances obtained between the vehicles (NHTSA, 2014).

Bai and Krishnan (2006) analyze the V2V communication reliability (i.e., DSRC wireless communication) by using driving assistance models when considering two real scenarios (traffic environment and freeway traffic environment). The results of the study presented by Bai and Krishnan show that the vehicular communication used gives an appropriate reliability level in the analysis of the average packet drop rate and packet drop pattern under various traffic environments, and from the viewpoint of the end-user.

In this contribution, the model is based on the work presented in (SEBASTIAN *et al.*, 2009), where the possibility of collision is calculated using vehicle kinematics based on the motion properties of the vehicle, such as trajectory and speed. This model is also based on several system assumptions, for example, it is assumed that all the vehicles are equipped with positioning, in-vehicle sensors and V2V communication systems. A GPS is used for the positioning system, and the in-vehicle sensors are used to obtain vehicle's heading, speed and acceleration. The noise influence is considered in the data from some of these devices. The V2V communication system is used to exchange messages of position coordinates, velocity, heading, acceleration, vehicle width and time-stamp between the vehicles using periodic broadcast so every vehicle can have up-to-date information from other vehicles. The Algorithm 1 provides the details of the *CW* procedure (this algorithm does not consider the curvature for the road used by the vehicles).

Through Algorithm 1 is possible to observe that the *CW* model is divided in two sections: *i*) vehicles in the same direction (*line* 2); and *ii*) vehicles in opposite direction (*line* 13). In the algorithm, it is also possible to identify three procedures in each one of the sections (just the first section has an additional procedure). These procedures allow us to find the needed parameters to determine a warning sign of a possible collision. In this model, not only we improved the model used as base, but we also proposed a new procedure to determine the Other vehicle (Ov) localization around Ego vehicle (Ev). Such location procedure allows the driver to be more specifically alerted about the Ov's direction in its environment.

Algorithm 1 – Collision Warning

```

1: for each pair of vehicles ( $Ev$ ,  $Ov$ ) do
2:   if  $|\theta_{Ev} - \theta_{Ov}| < \beta$  then
3:     compute the distances  $d_p$ ,  $d_{ls}$  and  $d_a$ 
4:     find the follower  $f$  and the leader  $l$ 
5:     compute following safe distance  $d_{sf}$ 
6:     find the location of  $Ov$  around  $Ev$ 
7:     if  $d_a < d_{sf}$  and  $d_p < d_{ls}$  and  $d_p < l_d$  then
8:       Collision warning
9:       Show the  $Ov$  location
10:    else if  $d_{ls} < d_p < l_d$  then
11:      Show the  $Ov$  location
12:    end if
13:  else if  $||\theta_{Ev} - \theta_{Ov}| - \pi| < \beta$  then
14:    compute the distances  $d_p$ ,  $d_{ls}$  and  $d_a$ 
15:    compute opposite safe distance  $d_{so}$ 
16:    find the location of  $Ov$  around  $Ev$ 
17:    if  $d_a < d_{so}$  and  $d_p < d_{ls}$  and  $d_p < l_d$  then
18:      Collision warning
19:      Show the  $Ov$  location
20:    else if  $d_{ls} < d_p < l_d$  then
21:      Show the  $Ov$  location
22:    end if
23:  end if
24: end for

```

4.2.1.1 Vehicles in the same direction

The first step to be taken is to determine if Ev and Ov are moving in the same direction. This could be done by comparing the direction angles between the vehicles, in which $\theta_{Ev} \approx \theta_{Ov}$. In Algorithm 1, this comparison is determined as $|\theta_{Ev} - \theta_{Ov}| < \beta$, in which β is a constant to be determined. Next, the distances d_a and d_p between the Ev and Ov vehicles have to be computed. These distances are represented in Figure 12 and may be calculated using equations Equation 4.10 and Equation 4.12. However, to get Equation 4.10 it is necessary to consider the value of the next variables:

$$\begin{aligned} x_{Ev'} &= \sin \theta_{Ev} + x_{Ev} \\ y_{Ev'} &= \cos \theta_{Ev} + y_{Ev} \end{aligned} \quad (4.6)$$

$$u = \frac{(x_{Ov} - x_{Ev})(x_{Ev'} - x_{Ev}) + (y_{Ov} - y_{Ev})(y_{Ev'} - y_{Ev})}{\sqrt{(x_{Ev'} - x_{Ev})^2 + (y_{Ev'} - y_{Ev})^2}} \quad (4.7)$$

$$x_P = x_{Ev} + u(x_{Ev'} - x_{Ev}) \quad (4.8)$$

Figure 12 – Vehicles with the same direction.

Source: Sebastian *et al.* (2009).

$$y_P = y_{Ev} + u(y_{Ev'} - y_{Ev}) \quad (4.9)$$

Using the Euclidean Distance is possible to find d_p :

$$d_p = \sqrt{(x_P - x_{Ov})^2 + (y_P - y_{Ov})^2} + \varphi(m, p) \quad (4.10)$$

In Equation 4.10, the $\varphi(m, p)$ represent the curve from the actual road in function of a map and the position of Ev and Ov . For this study the function $\varphi(m, p) = 0$, therefore, only the straight roads are considered in Equation 4.10. On the other hand, the compute lateral safety distance d_{ls} allows to decide if the two vehicles are in the same lane, considering the width of Ev (ω_{Ev}), the width of Ov (ω_{Ov}) and a minimal lateral safety distance d_{mls} that has to be defined.

$$d_{ls} = \frac{\omega_{Ev}}{2} + \frac{\omega_{Ov}}{2} + d_{mls} \quad (4.11)$$

The longitudinal distance d_a is also calculated based on the Euclidean Distance between the Ev position and the found coordinate P using Equations 4.6, 4.7, 4.8 and 4.9:

$$d_a = \sqrt{(x_P - x_{Ev})^2 + (y_P - y_{Ev})^2} \quad (4.12)$$

The next step is to find what is the role of Ev and Ov in the environment (i.e., a vehicle can have a role of *leader* or *follower*). In order to find these roles, it was necessary to modify the original model. In this new approach, it was defined the vector $\overrightarrow{EvP}$ between the points Ev

and P after it was found the angle ϕ between $\overrightarrow{EvP}$ and the velocity vector for Ev $\overrightarrow{V_{Ev}}$ using Equation 4.13:

$$\phi = \arccos \frac{\overrightarrow{EvP} \cdot \overrightarrow{V_{Ev}}}{|\overrightarrow{EvP}| |\overrightarrow{V_{Ev}}|} \quad (4.13)$$

If $\phi < \pi/2$ the Ev is a *follower* otherwise it is a *leader*. The next step is to calculate the safety distance d_{sf} using Equation 4.14:

$$d_{sf} = d_{min} + v_f t_r + \frac{1}{2} \left(\frac{v_f^2}{a_f} - \frac{v_l^2}{a_l} \right) \quad (4.14)$$

In which d_{min} is the minimal distance between Ev and Ov in order to avoid a collision; t_r is a parameter of driver's reaction time; v_f is the following vehicle speed; a_f is the following vehicle maximum deceleration; v_l is the leading vehicle speed; and a_l is the leading vehicle maximum deceleration. The value of the distances d_p , d_{ls} , d_a , d_{sf} and the value of the constant l_d (i.e., l_d is a constant that defines the maximum lateral distance for Ev) are necessary to determine the CW condition (Algorithm 1).

4.2.1.2 Location of Ov around Ev

In general, the CW systems report the risk possibility in some specific position (rear or front positions) depending on the number and type of sensors used. However, on some risk cases may require additional information for the driver (e.g., poor visibility or driver's distractions when changing lanes). In this regard, our proposal presents a new algorithm that visually informs the driver of Ov 's location around Ev . This new algorithm complements the model used as base in this thesis for the proposed cooperative collision warning system. Considering the information presented in Figure 12 the first part of the location algorithm consists in: *i*) determining the $\overrightarrow{EvP}$ and $\overrightarrow{EvOv}$ vectors; *ii*) defining the unit vector u in the y -axis direction; *iii*) obtaining the angle α between the $\overrightarrow{EvP}$ and u vector using Equation 4.13; *iv*) rotating an angle α , the vectors $\overrightarrow{EvP}$ and $\overrightarrow{EvOv}$ (henceforth will respectively be called $\overrightarrow{EvPr}$ and $\overrightarrow{EvOvr}$ referring to rotation). It is important to highlight that when the vector $\overrightarrow{EvOvr}$ is rotated it has to be in the first or second quadrant; therefore the rotation's direction has to be considered.

The above four steps allow to locate the vectors $\overrightarrow{EvP}$ and $\overrightarrow{EvOv}$ with respect to the positive y – axis. Such location allows determining a unique reference for the conditions that can be found between the vectors. From this reference, the second part of the algorithm considers the next steps to be taken: *i*) to get the angle β between the vectors $\overrightarrow{EvP}$ and $\overrightarrow{EvOv}$ using Equation 4.13; and *ii*) find the determinant of the matrix $D_{2 \times 2}$ from the rotated vectors $\overrightarrow{EvPr}$ and $\overrightarrow{EvOvr}$ (depending on the role of Ev and Ov - *leader* or *follower*). If Ev has a *follower* role, the

first matrix row D gets coordinates from vector $\overrightarrow{EvPr}$; otherwise, the first matrix row D gets the coordinates from vector $\overrightarrow{EvOvr}$. Equation 4.15 presents the matrix D when Ev is a *leader*.

$$D = \begin{pmatrix} EvOvr_x & EvOvr_y \\ EvPr_x & EvPr_y \end{pmatrix} \quad (4.15)$$

Finally, when the role of Ev , the angle and the determinant between the rotated vectors $\overrightarrow{EvPr}$ e $\overrightarrow{EvOvr}$ are known, it is possible to obtain the Ov location around Ev . In order for that to happen, the Ev environment needs to be divided in eight zones around it, where each zone presents a unique condition in function of the role, the determinant value, and the angle between the rotated vectors. Table 2 describes each zone and the necessary conditions of these three variables. For each of zones described in Table 2, the range of the angles has been determined by experimentation.

Table 2 – Zones around Ev

Zone	Role	Determinant	Angle
Front	follower	$\det = 0$	$\beta < 6.0$
Front Right	follower	$\det \leq 0$	$65 \geq \beta \geq 6.0$
Front Left	follower	$\det \geq 0$	$65 \geq \beta \geq 6.0$
Behind	Leader	$\det = 0$	$\beta < 6.0$
Behind Right	Leader	$\det \leq 0$	$65 \geq \beta \geq 6.0$
Behind Left	Leader	$\det \geq 0$	$65 \geq \beta \geq 6.0$
Right side	None	$\det \geq -11$	$90 \geq \beta \geq 65$
Left side	None	$\det \leq 11$	$90 \geq \beta \geq 65$

Source: Research data.

4.2.1.3 Vehicles in opposite directions

In this section, the Ev and Ov vehicles are considered to be in opposite directions if $||\theta_{Ev} - \theta_{Ov}| - \pi| < \beta$, in which β is the same constant used in subsection 4.2.1.1. Then, the lateral d_p , lateral safety d_{ls} , and longitudinal d_a distances should be calculated between the vehicles using Equations 4.10, 4.11 and 4.12. Opposite safe distance d_{so} is calculated using Equation 4.16.

$$d_{so} = d_{min} + v_{Ev}t_{rEv} + \frac{1}{2} \left(\frac{v_{Ev}^2}{a_{Ev}} \right) + v_{Ov}t_{rOv} + \frac{1}{2} \left(\frac{v_{Ov}^2}{a_{Ov}} \right) \quad (4.16)$$

The values of d_p , d_{ls} , d_a , d_{so} , and l_d determine a *CW* condition (Algorithm 1). Finally, the calculation of the location of Ov when in front of Ev is executed using the algorithm presented in subsection 4.2.1.2, considering the Ev vehicle in a *follower* role. Figure 13 represents every consideration required for two vehicles with opposite direction.

Figure 13 – Vehicles with opposite direction.

Source: Sebastian *et al.* (2009).

4.2.2 System design

In this section we describe the new version of the driving simulation platform presented in Chapter 3, for testing the developed model. The simulations were executed in a laptop Dell Vostro 5470, with processor Intel Core i7 2.0 GHz and 64 bit, memory of 8GB, video card NVIDIA Geforce with 2 GB, and operating system Ubuntu 14.04 Trusty Tahr.

4.2.2.1 V2V communications

The communication model is also implemented using the *NS-3* simulator (i.e., the version used was ns-3.19). On the other hand, into the code that executes the vehicular communication in *NS-3* it is also considered the creation of one component, which executes the communication between the software systems (*NS-3*, *MORSE* and *ROS*) using sockets. With this *socket* is possible to share the data contained in *NS-3* (*socketM-N* and *socketN-R* in Figure 14).

The basic requirements for the V2V communication network implemented in *NS-3* are: *i*) broadcasting communication (i.e., bidirectional) between all the *Ov* vehicles that interact with *Ev* in the simulation; *ii*) a variable time delay with respect to the distance between the *Ov* vehicles and *Ev*; *iii*) a communication range of 300 meters around *Ev*; and *iv*) a data rate of 6Mbps for a 10MHz bandwidth. One node in *NS-3* represents one vehicle in the simulation. Moreover, each node is set with all the characteristics from a vehicular network (i.e., application, channel, net devices and protocol) (LACAGE; BU, 2013). Finally, in the V2V communication implemented, ten *Ov* vehicles share messages with *Ev* vehicles, depending on the communication coverage determined.

The second requirement for the V2V communication network is based on the study presented by Teixeira *et al.* (2014). Their proposal measures the performance in a V2V communication between two vehicles, using one laptop for each vehicle with a communication card DCMA86 – P2. Each communication card works with the standard 802.11p. The results obtained by Teixeira *et al.* show how the time delay in the V2V communication could have effects on a driver assistance system. Therefore, in the V2V communication model used in this proposal, the time delay effect is added, taken as base the results obtained by Teixeira *et al.* in a real test field (i.e., this proposal considered the values from time delay in the V2V communication with respect to the distance between the vehicles with a constant velocity determined).

4.2.2.2 Vehicle simulation

The driving simulation platform presented in Chapter 3 was implemented using, as the main tool, the Modular Open Robots Simulation Engine *MORSE* (i.e., the version used in this thesis was morse 1.2.2). The simulations in *MORSE* are based in *Blender* (i.e., the version used in this thesis was blender-2.73a). This characteristic in *MORSE* allows, in this proposal, the creation of different 3D-vehicles models, one 3D-highway scenario, one speedometer and one coalition warning indicator in *Ev*.

In this contribution, it was also used the Robot Operating System *ROS* (QUIGLEY *et al.*, 2009) (i.e. the version used in this thesis was Indigo). The *ROS* application in this proposal can be divided in three main parts: *i*) getting messages from the simulation in *NS – 3* (using the *socketN – R*, Figure 14) of the *Ov* vehicles that are in the communication coverage determined for *Ev*; *ii*) applying the model proposed in subsection 4.2.1, considering the messages obtained in the previous part, the data obtained from the sensors and the discrete Kalman filter presented in sub section 4.2.2.3 from *Ev*; and *iii*) sending the messages from *Ev* in *ROS* to the simulation in *NS – 3* using a *socketR – N* (Figure 14).

4.2.2.3 Vehicle position estimation

For the *Ev*'s location in the simulation environment it was used a GPS sensor of *MORSE*, which provides the Cartesian coordinates for *Ev* in the simulation scenario. With the aim to get a more real data read from the GPS it was used a modifier in *MORSE*, which added a Gaussian noise on the data obtained by the GPS. The Gaussian noise added had a standard deviation of 0.5 meters. To get an accuracy level on *Ev*'s location it was developed a Discrete Kalman Filter (*DKF*) based on (WELCH; BISHOP, 2006), in order to integrate the GPS data with a velocity sensor of *MORSE* on *Ev* (the velocity sensor in *Morse* provides data about the linear and angular velocity of *Ev*). Furthermore, a pose sensor of *MORSE* was also used to calculate *Ev*'s direction in the simulation scenario.

For the *DKF* it was determined the following conditions for the *Ev* vehicle: *i*) displacements in only two dimensions (x and y); *ii*) an initial direction (ψ); *iii*) a linear velocity (v);

displacement in a highway scenario. Table 3 presents the statistical description of the distance between all the points from the *Real* signals (line blue) with respect to the *GPS* (line yellow) and *DKF* (line green) in Figure 15.

Figure 15 – Result of *DKF* application in the *Ev* displacement

Source: Elaborated by the author.

Table 3 – Comparison between the signals from figure 15

Distance between points	μ (m)	σ (m)	σ^2 (m)
Real-GPS	0.672681	0.350979	0.123186
Real-DKF	0.209613	0.084525	0.007144

Source: Research data.

With the results presented in Table 3, it is possible to see that the output from *DKF* assures a superior accuracy level in the *Ev*'s location, and therefore a better determination from risk situations using Algorithm 1.

4.2.3 Simulation results

In this subsection, the tests and results obtained from applying the cooperative model for the proposed driving assistance are described. In the experiments were taken into consideration the indications presented in Annex D.3 in (KNAPP *et al.*, 2009), and the use of the system presented in subsection 4.2.2. In the first developmental stage, when assessing the system, it was implemented a self-driving algorithm in all the vehicles (i.e., this algorithm is used by *Ev*). This algorithm has the objective to execute the driver's activities, within the scenario, during the data collection⁵. In each tests, it was computed the value of the safety distance d_{sf}

⁵ This algorithm was developed by Tiago Cesar dos Santos in the Mobile Robotics Laboratory, ICMC-USP, Brazil.

(i.e., or d_{so}) and the longitudinal distance d_a during a warning collision signal (this signal is represented in the simulation by a red arrow and a sound). Also, the coverage ranges (maximum and minimum) from the warning signals activated within the zones around E_v were determined; they are presented in Table 2 (these signals are represented in the simulation by a green arrow).

The simulation scenario created to calculate these computations was based on a specific highway segment of the Washington Luiz Highway (without considering its real geometric characteristics) in São Paulo state, Brazil (from the bridge at the main access way to São Carlos Street until the u-turn at exit 240). This scenario has two specific transit types: one-way transit and two-way transit; and three types of collision situations: lead vehicle moving at lower constant speed, following vehicle making a maneuver and vehicle(s) on opposite directions. These collision situations were proposed in (HARDING *et al.*, 2014a). The simulation was set to execute for one hour for each collision situation. A video with an example of each experiment is presented in <<https://youtu.be/UP-FCKLGRSU>>.

4.2.3.1 Experiments

One-way transit

In this experiment are considered two types of collision situations: *i*) lead vehicle moving at lower constant speed, and *ii*) following vehicle making a maneuver. In the first collision situation, the E_v vehicle interacts with other ten O_v vehicles within the highway scenario. The O_v vehicles are distributed throughout the simulation scenario on the right lane of the highway, and they are moving with a constant velocity of 36 km/h. The E_v vehicle is also moving on the right lane with a velocity of 60Km/h. All vehicles are sending messages continually.

When the E_v vehicle is found within the communication coverage with one or more O_v vehicles, a warning signal is turned on (green arrow). This warning signal shows the zone around the E_v vehicle where the system detected one O_v vehicle. On the other hand, when the E_v vehicle and an O_v vehicle are in compliance with the conditions presented in lines 2 and 7 of Algorithm 1 a signal of warning collision is turn on (red arrow).

After a few seconds, the self-driving algorithm that executes the driver's activities in the vehicles, makes the O_v vehicle in front of the E_v vehicle change lanes, and increases its velocity when the O_v vehicle moved onto the left lane (here, the vehicles on the left lane are faster that the vehicles on the right lane). This procedure is repeated throughout the experiment with each one of the ten O_v vehicles within the scenario. Added to that, in two specific locations of the scenario, the O_v vehicles return to the right lane and decrease their velocity, so that O_v vehicles meet the E_v vehicle again at another moment of the simulation (the E_v vehicle does not change lanes throughout the experiment).

For the second collision situation are used all the characteristics from the first collision situation, with the only difference being that the E_v vehicle moves with a velocity of 36Km/h

(once again on the right lane) and the Ov vehicles with a velocity of $60Km/h$.

Two-way transit

In this experiment the collision situation used is a vehicle(s) not making a maneuver on the opposite direction. Here, the Ev vehicle is with ten Ov vehicles, but on the left lane. All the vehicles have the same velocity ($60Km/h$). The vehicles in the simulation are separated from each other by $470m$. Once again all vehicles are sending messages constantly.

During the experiment, the warning signals always turn on within the communication coverage $V2V$ (green arrows). When the Ev vehicle is opposite to an Ov vehicle on the same lane, and the conditions presented in lines 13 and 17 of the [Algorithm 1](#) are complied, a collision warning signal turns on (red arrow). A few seconds after, the Ov vehicles change lanes and increase their velocity. Here, in two specific locations of the scenario, the Ov vehicles return to the left lane and decrease their velocity.

4.2.3.2 Results

Considering the large amount of data recorded in each collision situation, it was selected one segment of time for each experiment described in subsection [4.2.3.1](#). Part of the data of these segments are presented in the [Figure 16](#), in which the main idea is to present the relation between the Ev vehicle and the Ov vehicles in each experiment. This relation was determined taking into consideration the longitudinal d_a and safety distances (i.e., d_{sf} in subsection [4.2.1.1](#) or d_{so} in subsection [4.2.1.3](#)) computed by the system. In [Figure 16](#), the sub-figures represent the results regarding the first risk instant, that is, when the proposed system warns, by the first time, about a possible collision between the Ev vehicle and one of the Ov vehicles in the simulation. For an easier understanding, it is highlighted in red the point that represents the safety distance computed by the system (Equations [4.14](#) and [4.16](#)). The other points and triangles with different colors represent the longitudinal distance between the Ev vehicle and an Ov vehicle during the event. Given the large amount of data collected in the experiments, the graphical representation of the results is complex. For this reason, we considered important to stand out that in each sub-figure the data for the warning signals around the Ev vehicle (green arrows) and the sequence of the collision warning signals (i.e., red arrows), with respect to each Ov vehicle, was filtered.

In the sub-figures, it is also possible to see ranges for the longitudinal and safety distances. These ranges were determined considering the values from the longitudinal and safety distances during the first collision warning, throughout the entire time interval of the experiment's execution. The maximum and minimum values in each range were determined by applying a bootstrap distribution, with a confidence interval of 95% for each type of distance (d_{sf} or d_{so} and d_a). For each bootstrap distribution computed, were used the mean values μ_v of the longitudinal and safety distances as samples for the distributions (the mean values μ_v were obtained from the recorded values for d_{sf} or d_{so} , and d_a during the first collision warning, for each vehicle, in all

the experiment). In [Figure 16c](#), all the red points are outside the range safety distance, because the bootstrap distribution of this collision situation is located in the middle of all the safety distance values obtained throughout all the experiment.

In [Figure 16](#), for each collision situation, it is possible to see the irregular difference between the longitudinal and safety distances during a collision warning with respect to different Ov vehicles. The differences between the distances were resulted from the presence of other vehicles inside the communication coverage of the Ev vehicle during a collision warning event. Therefore, when each Ov vehicle sent a message for the Ev vehicle the time delay among them was different (that depends from the distances of each Ov vehicle regarding the Ev vehicle). Moreover, the usage order of the only communication channel produces the losses of some messages during a variable time delay period (that depends on the number of Ov vehicles that are within the communication coverage). This behavior, suggested by [Teixeira et al. \(2014\)](#), compromises performance of the proposed system.

With the objective to discover which of the three collision situations was the most affected by message losses during the first collision warning signal, it was computed the difference between the longitudinal and safety distances (d_{sf} or d_{so} and d_a) for each collision situation, regarding each Ov vehicle throughout the execution time of the experiments. Afterward, the mean values μ_{dif} were obtained for the differences of the distances in each collision situation, between the Ev vehicle and one Ov vehicle. In total, were obtained ten mean values (one for each Ov vehicle) from the differences, in each collision situation, which were used again as samples for a bootstrap distribution with a confidence interval of 95%.

The maximum and minimum values, obtained in each one of the three distributions, allowed to see that the third collision situation (vehicle(s) not making a maneuver with opposite direction) was where the performance of the proposed system presented the largest difference between d_{so} and d_a , with a maximum value in the bootstrap distribution of 24.14m, and the minimum of 13.39m. On the other hand, the second collision situation (following vehicle making a maneuver), had the smallest difference between d_{sf} and d_a , with a maximum value in the bootstrap distribution of 4.43m, and minimum of 2.43m. [Table 4](#) presents the coverage ranges obtained for each one of the warning signals used in the experiments. The coverage determination of each warning signal was considered by the calculation of the mean value μ and the standard deviation σ of the maximum and minimum distances for each warning signal, by Ov vehicle, in every experiment (the experiments are identified as CS).

(a) Lead vehicle moving at lower constant speed.

(b) Following vehicle making a maneuver.

(c) Vehicle(s) not making a maneuver with opposite direction.

Figure 16 – Results of d_a measures for each collision situation

Source: Elaborated by the author.

Table 4 – Range detection distance of the warning signals by zone around Ev

Zone Ev	Front		Front Right		Front Left		Behind		Behind Left		Left side	
CS-1	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Max (m)	292.49	6.65	None	None	61.38	5.87	242.33	67.60	50.04	7.14	1.76	0.27
Min (m)	36.97	4.12	None	None	3.48	0.54	45.08	3.19	3.28	0.93	0.47	0.33
CS-2	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Max (m)	288.85	14.86	None	None	53.64	1.99	301.18	2.50	46.76	7.68	1.39	0.53
Min (m)	50.31	6.74	None	None	3.11	1.62	43.80	4.00	3.19	0.94	0.41	0.37
CS-3	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Max (m)	295.89	0.73	29.51	5.53	None	None	None	None	None	None	None	None
Min (m)	46.43	5.82	4.47	1.66	None	None	None	None	None	None	None	None

Source: Research data.

4.3 Final considerations

In this chapter we have presented two applications of the driving simulation platform in the development and evaluation of cooperative vehicle systems. In each of the systems, a specific configuration was established in the driving simulation platform from the architecture presented in [Chapter 3](#). The established configurations were considered due to the nature of the experiments performed for each of the systems. In the development of the two systems presented, it was possible to observe an evolution in the original architecture of the driving simulation platform. The main difference is the new communication model used, including the creation of a road scenario and the interaction of the Ego-vehicle with a larger number of vehicles in the simulations. In the evaluation of the cooperative systems, we verified how the strict requirements of quality in the vehicular communication can affect their performance. Finally, the kinematic model of the system presented in [section 4.2](#) is the basic structure of the cooperative driver assistance system developed in the following chapter. Its implementation, improvement, and evaluation were important for the next phase of this thesis.

APPLICATION OF BAYESIAN NETWORKS ON COOPERATIVE DRIVING ASSISTANCE SYSTEMS

Just like the many tasks that we carry out in everyday life driving a vehicle also requires attention and reasoning on the part of the driver. To accomplish this specific task the driver should be able to obtain some information from his environment in order to decide how to act. However, the obtaining of this information by the drivers may be limited, in some situations, by our own human nature. This can lead to restricted decision-making, given the limited capacity of the human being to interpret part of their environment. In this sense, therefore, the driver assistance systems are configured as pressing need, since they can offer a complete reading of the environment for the driver, reduce uncertainties and avoid risky situations.

According to [Koller and Friedman \(2009\)](#), the uncertainty appears to be an inescapable aspect of most real-world applications. Therefore, it is a consequence of several factors. They point out that due to this omnipresent and fundamental uncertainty about the true state of the world, systems have to be modeled considering different possibilities. For this reason, the models must be probabilistic, since they are more real and exact.

To simplify the uncertainty situations it is possible to assume that some events exist. These events are clustered around variables, each with a set of outcomes, also called states. Mathematically, the structure is called a directed graph ([NIELSEN; JENSEN, 2009](#)). This technique in machine learning is known as Probabilistic Graphical Models (PGM). In this technique, it is possible to use a probability distribution on different variables to obtain a model ([KOLLER; FRIEDMAN, 2009](#)).

In this universe of uncertain situations, lane change was the event selected for analysis in this thesis. To solve the problem of lane change, the procedure used with the *PGM* was: i) to represent, ii) to construct, iii) to obtain structure and parameters, and iv) to make inferences

about queries. In this chapter, therefore, we describe in detail how this methodology was used and evaluated to develop a prototype of cooperative driver assistance system for lane change.

5.1 Bayesian Networks

Bayesian Networks (BN) is a graphical model with which it is possible to represent the uncertainty of a phenomenon, applying Bayesian statistics. Considering [Russell and Norvig \(2010\)](#), a Bayesian network is a Directed Acyclic Graph (DAG), in which each node has quantified probabilistic information. Some of the characteristics of Bayesian networks are:

- Each node corresponds to a random variable, which may be *discrete* or *continuous*;
- A set of directed links or arrows connects pairs of nodes. If there is an arrow from node X to node Y , X is said to be a parent of Y . The graph has no directed cycles (and hence is a directed acyclic graph); and
- Each node X_i has a conditional probability distribution $P(X_i \mid \text{Parents}(X_i))$ that quantifies the effect of the parents on the node.

Figure 17 – Example of a Bayesian network.

Source: [Koller and Friedman \(2009\)](#).

To illustrate, in [Figure 17](#) we indicate a Bayesian network structure that presents a model with a set of variables. Each of the variables has a Conditional Probability Distribution (CPD) assigned that makes it depend only on its corresponding relative. This topological order of the structure in the BN 's is its main advantage because it determines the dependency only with its parents and the conditional independence between the variables of the model. With this aspect of independence the number of conditional probabilities in the model decreases. In

(MURPHY, 2012), Murphy explains that this behavior of Bayesian networks is a function of the execution time of the algorithm. For Murphy, assuming that a Bayesian network is conditionally independent, and each node in the network has $O(F)$ parents and K states, the number of parameters in the model is $O(VK^F)$ (i.e., where V is the number of vertices). This is much less than the $O(K^V)$ needed by a model which makes no conditional independence assumptions.

In general, the conditional independence of variables in the structure of Bayesian networks is represented as a product of factors. Equation 5.1 formally represents this idea:

$$p(x_1, \dots, x_V) = \prod_{t=1}^V p(x_t | \mathbf{x}_{pa(t)}) \quad (5.1)$$

Equation 5.1 is known as *the chain rule for the Bayesian networks*, where x_t is a node in the graph G and $\mathbf{x}_{pa(t)}$ is the parents of the x_t node in the same graph G . In Equation 5.1, each term $p(x_t | \mathbf{x}_{pa(t)})$ is a *CPD*.

Finally, considering the study presented by Schubert (2012), Bayesian networks are a flexible graphical model that can adapt to the characteristics of a system. In addition, Bayesian networks have the ability to perform probabilistic rationalization, based on a formal concept that allows an efficient computational evaluation.

5.1.1 Building models

When choosing to create a model using Bayesian networks, the first step in constructing the model is to identify the events that may or may not be observable in the problem being treated. These events can be grouped into hypothesis variables (NIELSEN; JENSEN, 2009), which organize the events so that they have an exclusive relationship between them. After identifying the events, the hypothesis variables need information that allows predicting some characteristic of these variables. The second step, therefore, is to collect this information. The types of information required can also be stored in random variables, which store the information in a number of states. Similar to the hypothesis variable, these states should be mutually exclusive.

In a formal definition of random variables, uppercase letters are used for variables and lowercase letters for states. Random variables can be of two types: discrete or continuous. The main difference between these two types of variables is the number of possible states that can be found in these variables. We can exemplify a discrete random variable by representing the possible states to obtain the face of a coin. On the other hand, the velocity representation of a car can be an example of a continuous random variable. In this thesis was established not to work with the representation of continuous variables. We considered, therefore, the transformation of continuous variables into discrete variables by establishing intervals, as it is suggested by Nielsen and Jensen (2009) and Sucar (2015).

Having identified the variables for the model, the next step will be to establish the

directed links among themselves. Learning the network structure of the model is one of the most challenging aspects of building a Bayesian network (MARGARITIS, 2003). According to Sucar (2015), the learning of the structure of a Bayesian network will depend on a dataset with a considerable number of observations of the random variables chosen for the model. This aspect can be a complex problem since the number of possible structures is wide, even with a small group of variables.

According to several studies (ANKAN; PANDA, 2015; KARKERA, 2014; TRABELSI, 2013; NIELSEN; JENSEN, 2009; MARGARITIS, 2003) it is possible to find two common methodologies among the algorithms used to learn the structure of a Bayesian network. These two methodologies are: *i*) Constraint-based structure learning; and *ii*) Score-based learning. In the first methodology, the objective is to identify the independence between the variables defined through the observations obtained in the dataset. This is usually done using a statistical hypothesis test (TRABELSI, 2013). In the second methodology, a score is assigned to a possible structure indicating how much this structure conforms to the observations of the dataset. Then, within the set of all possible structures, it is wanted the best score (KARKERA, 2014).

According to Trabelsi (2013), although the first methodology presented earlier is more efficient, it is prone to have errors in the tests of independence. The second methodology, having its focus on optimizing the structure of the network, may be a better option. From this point, it is evident the complexity of the construction of a model using Bayesian networks. Fortunately, there are currently a lot of software tools (i.e., commercial and open-source) that allow the application of PGM's. For a more formal and practical learning of the Bayesian network structure, we recommend consulting the books by Ankan and Panda (2015), and by Karkera (2014).

Once the variables definition and Bayesian network topology are determined, the next step is to define the conditional probability distribution of each state of the variables in the model. This distribution task occurs in the same way as learning the Bayesian network structure, that is, from the dataset. The learning of these parameters can usually be done using two approaches (BELLOT, 2016; ANKAN; PANDA, 2015; KARKERA, 2014): *i*) the maximum likelihood estimation, and *ii*) Bayesian statistics.

According to Bellot (2016), D is a dataset and θ is a parameter of our Bayesian network. Using these two assignments, the probability function $P(D|\theta)$ can be defined as the probability of generating the dataset, according to the parameters. The maximum likelihood estimation approach identifies the value of the parameter θ , which maximizes the probability function $P(D|\theta)$. This can be defined as $\tilde{\theta} = \operatorname{argmax}_{\theta} P(D|\theta)$. While the Bayesian approach provides a prior distribution of the parameters θ , $P(\theta)$. In this case, finding the parameter value boils down to finding the maximum value for $P(D|\theta).P(\theta)$. This is called a *maximum a posteriori*.

5.1.2 Inference

Whenever the Bayesian network is completed the inference task of this machine learning technique can be accomplished. The inference consists in propagating the effects of certain evidence in a Bayesian network to estimate its effect on the unobserved variables (SUCAR, 2015). In other words, the inference in Bayesian networks tries to find the conditional probability of an unobserved variable, given the values, observed in other variables of the built Bayesian network.

According to some studies (BELLOT, 2016; ANKAN; PANDA, 2015; SUCAR, 2015; KARKERA, 2014), this estimation of an unobserved variable can be done through two types of queries: *i*) probabilistic queries; and *ii*) Maximum a Posteriori Assignment (MAP) queries. Specifically, in the first type of query, the main interest is the value of the marginal probability of the variable not observed in the model. In the second type of query, the main interest is to obtain the state of the variable not observed in the model.

The problem of inference in graphical models is $NP - hard$, and therefore it probably requires exponential time in the worst case (except in the unlikely event that $P = NP$) (KOLLER; FRIEDMAN, 2009). However, in many real-world applications, this problem can be effectively addressed using some inference algorithms (i.e., exact or inaccurate inference). Two of the most referenced algorithms in the state of the art of PGM 's are (NIELSEN; JENSEN, 2009; KOLLER; FRIEDMAN, 2009; MURPHY, 2012; SCUTARI; DENIS, 2014; KARKERA, 2014; ANKAN; PANDA, 2015; SUCAR, 2015; BELLOT, 2016): *i*) variable elimination; and *ii*) Junction tree.

5.2 Dynamical Bayesian Networks

Considering Russell and Norvig (2010), a real-world problem can be modeled as a sequence of temporal fractions. These temporal fractions may have a set of random variables, from which it is possible to extract an observation from each temporal fraction (Figure 18). The next step in modeling the problem is to specify a relationship between each of these time fractions and their respective set of random variables. Taking into account that the interval of temporal fractions depends on the type of problem being treated, the valuation of the model to be established depends on the relation between time and uncertainty.

This type of model can be developed taking into account two main assumptions (MURPHY, 2002; KOLLER; FRIEDMAN, 2009; SUCAR, 2015):

- First-order Markov model: the state variables at time t depend only on the state variables at time $t - 1$ (and other variables at time t); and
- Stationary process: the structure and parameters of the model do not change over time.

Figure 18 – Diagram showing a model where time slices are used to represent a temporal process.

Source: Mihajlovic and Petkovic (2001).

All previous specifications for the modeling of a temporal problem can be found in a Dynamic Bayesian Network (DBN). Since we only consider processes that are stationary and Markovian, and since the observations at a given time are only dependent on the state at that time, two consecutive time slices are sufficient to graphically depict the dynamic model and observation models using the Bayesian network machinery (GIMPEL; RUDOY, 2008). In this perspective, therefore, a *DBN* can be formally defined as an ordered pair (B_O, B_t) , where B_O is a Bayesian network representing the prior distribution $p(X_o)$ of a process and B_t is a 2-time slice Bayesian network ($2 - TBN$) (Figure 19) which defines the dynamic model $p(X_t|X_{t-1})$ of the process (KOLLER; FRIEDMAN, 2009; GIMPEL; RUDOY, 2008). Using these models, a *DBN* provides a compact factorization for the posterior distribution $p(X_{0:T})$ for any T :

$$p(X_{0:T}) = p(X_0) \prod_{t=1}^T p(X_t|X_{t-1}) \quad (5.2)$$

Note that the term *dynamic* means we are modeling a dynamic system, not that the network changes over time (MURPHY, 2002).

5.2.1 Building models

Both *BN's* and *DBN's* should consider the same three main aspects for the construction of a model: *i*) problem variables; *ii*) topology learning; and *iii*) learning the parameters (i.e., CDP's) for each variable. According to Santos (2017) and Sucar (2015), for each of these aspects in the *DBNs*, we can apply the same methods used in the *BN's* mentioned in subsection 5.1.1.

Considering the definition of a *DBN* presented in section 5.2, learning of the structure and the parameters of B_t (i.e., or $2 - TBN$) can be developed considering the temporal information of the data of each of the variables in two consecutive fractions of time, X_{t-1} e X_t (SUCAR, 2015). This whole process can be accomplished by the two main assumptions mentioned in section 5.2.

Figure 19 – A 2-time slice Bayesian network (2 – TBN), which shows the dynamic and observation models using two consecutive time slices of a process

Source: [Gimpel and Rudoy \(2008\)](#).

5.2.2 Inference

With the construction of the *DBN* model there is the possibility of performing some classes of inference. Among the best-known inference classes are ([SUCAR, 2015](#); [RUSSELL; NORVIG, 2010](#)):

- **Filtering:** it is the task of computing the belief state (the posterior distribution over the most recent state) given all evidence to date;
- **Prediction:** it is the task of computing the posterior distribution over the future state, given all evidence to date;
- **Smoothing:** it is the task of computing the posterior distribution over a past state, given all evidence up to the present; and
- **Most likely explanation:** given a sequence of observations, we might wish to find the sequence of states that is most likely to have generated those observations.

5.3 Driver assistance system for lane change

Human errors are the main factor that justifies the high numbers of traffic accident mortality ([SINGH, 2015](#)). Among the most common examples of these errors are: *i*) cell phone use at the time of driving; *ii*) abuse of alcohol and other narcotic substances; and *iii*) lack of tolerance. In addition to these harmful actions, there are also natural aspects of the human being that can contribute to the increase of risk in the direction, including the reduction of skills when the person reaches the elderly phase of life.

Considering [Abendroth and Bruder \(2016\)](#), a vehicle guidance is a predominantly informational activity in which work content and information are converted into reactions. A

driver is usually processing environmental information based on his perception (i.e., most of the information relevant to driving an automobile is taken in visually - usually around 80–90% (ROCKWELL, 1972)). This perception of the environment is influenced by situations that can happen in the environment and for their experience in the direction. In this perspective, the decisions at the moment of driving a vehicle can be influenced by characteristics of each driver.

Through these aspects presented, Driver Assistance Systems are one of the alternatives that can help to reduce this problem of accidents on the roads. In this sense, the main purpose of a Driver Assistance System is to support drivers in the direction of a vehicle. This system has an implicit interaction with human drivers and its use may be potentially beneficial with respect to the tasks of steering a vehicle. The lane change maneuver is among the actions that have a high error potential. More than 5% of all accidents involving injury to people take place as the result of a lane change (BARTELS; MEINECKE; STEINMEYER, 2016).

For a lane change maneuver to be executed two important aspects must be considered (NILSSON *et al.*, 2016): *i*) that the maneuver is possible; and *ii*) that the maneuver is desirable. The lane change maneuver is possible if it avoids collision conflicts with all surrounding vehicles; respects traffic rules and regulations; and satisfies physical and design limitations. Nilsson complements pointing out that the lane change maneuver is desirable, when you want to improve conditions in the direction of the vehicle (e.g., in order to allow a fast trailing vehicle to pass); or when the track conditions require it (e.g., due to a lane drop).

Currently, you can find different proposals to help the driver in the specific task of lane change, including the study of Amodio, Panzani and Savaresi (2017). These authors propose a passive system of assistance to the motorcycle driver in order to only inform when it is possible to make a lane change without the need to use the motorcycle mirrors. To achieve this goal, they use an instrumented motorcycle with a short-range radar sensor mounted to the rear of the vehicle and a system of LEDs to interface with the driver.

We can cite another example used by Lee, Hansen and Hedrick (2017) to address the problem of the lane change. The authors of this proposal, incorporated in an adaptive cruise control system, a model that determines the intention to lane change of the vehicles nearby. The main objective of this study is to increase the safety and comfort of drivers with the inclusion of this model. To obtain this model, the authors used graphics modeling techniques and unsupervised learning. Exact probabilistic inference is used to obtain a posterior probability distribution over the driver's intention. A radar and a camera were used as sensors in this proposal.

In (DANG *et al.*, 2014), the assistance proposal for lane change is based on a V2V communications system. This proposal consists of four components, which formally present a deterministic model. To determine a possible lane change maneuver, the model compares the actual distance between vehicles with a minimum safety distance established. The evaluation of the model was made using Simulink as the main tool.

Bartels, Meinecke and Steinmeyer (2016) establish the following functional requirements for lane change assistance systems: *i)* the lane change assistant should inform the driver about hazardous situations that result due to the driver failing to monitor the area around the vehicle adequately; *ii)* the assistance function should be capable of detecting road users approaching rapidly from behind as well as detecting other road users in the blind spot of the subject vehicle; *iii)* the assistance function should operate for both adjacent lanes equally, on the driver and the passenger sides; and *iv)* ideally, the assistance function will be available for all road, weather, and traffic conditions with approximately the same level of quality.

In (NILSSON *et al.*, 2016), the authors consider the problem of establishing *if*, *when*, and *how* to perform a lane change maneuver. The algorithm used by the authors considers the problem of lane change in three aspects: *i)* decide whether a lane change maneuver is desirable; *ii)* select the inter-vehicle traffic gap and time instance to perform the maneuver; and *iii)* plan the longitudinal and lateral control trajectory to execute the maneuver. The decision regarding the feasibility and execution of the maneuver is formulated as an optimal control trajectory planning problem. The results of the proposal are obtained through simulation and tests in a real track.

Luo *et al.* (2016), propose a dynamic automated lane change maneuver based on vehicle-to-vehicle communication. With this method, it is possible to plan a reference route to perform a lane change during a normal or emergency situation. The proposed dynamic automated lane change maneuver consists of two parts: *i)* trajectory planning; and *ii)* trajectory tracking. In the first part, all information is obtained through sensors and communication V2V. In the second part, the maneuver controls the system based on the tracking errors between the current trajectory and the reference trajectory. All results are obtained by simulated experiments.

In (SIVARAMAN; TRIVEDI, 2014), the authors present the dynamic probabilistic drivability map (DPDM). The DPDM is a data structure that contains spatial information, dynamics, and probabilities of drivability; readily integrating measurements from a variety of sensors. Using the DPDM, the authors develop a general predictive system for lane change and merge. The full system provides timing and acceleration recommendations, which are designed to advise the driver *when* and *how* to merge and change lanes. The Lane change and merge system has been extensively tested using real-world on-road data from urban driving, dense highway traffic, free-flow highway traffic, and merge scenarios.

Nilsson, Fredriksson and Coelingh (2015), propose an algorithm based on the formulation of simple logical rules to recognize the intent of a lane change maneuver on a highway. The authors consider that the rule-based systems have the advantage of traceability and ease of implementation, and provide a clear overview and understanding of the system model. The system is based on rules proposed by the authors and takes into account suggestions of lateral and longitudinal movements to recognize the intention of the lane change maneuver. The results of the evaluation of the proposed system were obtained using a simulator.

In (BUTAKOV; IOANNOU, 2015), the authors consider that driver assistance systems

are developed in a standard way, without taking into account the particular characteristics of each driver. These authors propose, therefore, a modeling framework that takes into account the driver/vehicle response before and during the maneuver that can be utilized by ADAS to provide personalized assistance to drivers. The proposed model consists of two layers: *i*) a stochastic model; *ii*) and a kinematic model. This proposal was evaluated in a real test field.

Finally, Yan, Weber and Luedtke (2015) study the driver's uncertainty about distance gaps and its impact at lane changing. The authors assume that a reduction of driver's uncertainty with assistance systems during lane change situations can not only increase traffic safety, but also increase driver's trust in assistance systems. In this study it was made a driving simulator experiment to classify driver's uncertainty about the distance gap during lane changing.

5.4 Cooperative driver assistance system for lane change

Considering the context presented in the previous section, it was possible to highlight three important aspects of the development of a driver assistance system: *i*) specific definition of the type of assistance that the driver of a vehicle will receive; *ii*) construction of a model that represents the defined assistance type to the driver; and *iii*) evaluation of the developed model. Taking into account these three aspects, we propose in this contribution a cooperative driver assistance system for the lane change which can help avoid collisions among the closest vehicles, thus making the lane change maneuver safe. This is possible by informing some suggestions to the driver. The model developed for this purpose is a new version of the model used in section 4.2. The new model proposed in this research is the combination of a kinematic model with a probabilistic graphical model (PGM). Finally, this proposal will be evaluated in a new version of the simulation platform presented in Chapters 3 and 4.

5.4.1 Kinematic model

The new version of the kinematic model used in this contribution is based on the study presented by Sebastian *et al.* (2009). In this new version of the kinematic model, again the determination of a situation of risk between the vehicles is once again given by the sharing of kinematic information of each vehicle. This is possible by implementing a V2V communication model. In this model, therefore, we assume that all vehicles have positioning, in-vehicle sensors and V2V communication systems. Similar to the model presented in section 4.2, this approach also considers the influence of noise on the data acquired by the sensors. To correct the noise effect, the same discrete Kalman filter shown in subsection 4.2.2.3 was used. Algorithm 2 provides the details of a lane change procedure.

In relation to the kinematic model presented in subsection 4.2.1, the kinematic model shown in Algorithm 2 has three specific novelties: *i*) inclusion of the vehicle's steering angle value (lines 7,19 and 29); *ii*) a new condition with regard to the direction of vehicles (line 27);

Algorithm 2 – Lane change

```

1: for pair of vehicles ( $E_v, O_v$ ) do
2:   if  $|\theta_{E_v} - \theta_{O_v}| < \beta$  then
3:     compute the distances  $d_p$ ,  $d_{ls}$  and  $d_a$ 
4:     find the follower  $f$  and the leader  $l$ 
5:     compute following safe distance  $d_{sf}$ 
6:     compute the lane of  $O_v$ 
7:     if  $d_a < d_{sf}$  and  $St_{E_v} < St_k$  and  $St_{O_v} < St_k$  and  $d_p < l_d$  then
8:       find the location of  $O_v$  around  $E_v$ 
9:       execute class critical( $scan, constans, vectors, t, id$ )
10:      if  $scan = 5$  then
11:        execute class inference( $L_{st}, Vd_{st}, Lc_{st}$ )
12:        information for a driver
13:      end if
14:    end if
15:    else if  $||\theta_{E_v} - \theta_{O_v}| - \pi| < \beta$  then
16:      compute the distances  $d_p$ ,  $d_{ls}$  and  $d_a$ 
17:      compute opposite safe distance  $d_{so}$ 
18:      compute the lane of  $O_v$ 
19:      if  $d_a < d_{so}$  and  $St_{E_v} < St_k$  and  $St_{O_v} < St_k$  and  $d_p < l_d$  then
20:        find the location of  $O_v$  around  $E_v$ 
21:        execute class critical( $scan, constans, vectors, t, id$ )
22:        if  $scan = 5$  then
23:          execute class inference( $L_{st}, Vd_{st}, Lc_{st}$ )
24:          information for a driver
25:        end if
26:      end if
27:    else
28:      Compute Ackerman's radius  $r_{ack}$ , curve distance  $d_{sc}$  and distance  $d_{EO_v}$ 
29:      if  $d_{EO_v} < d_{sc}$  and  $St_{E_v} > St_k$  then
30:        caution alarm
31:      else if  $d_{EO_v} < d_{critical}$  then
32:        caution alarm
33:      end if
34:    end if
35: end for

```

and *iii*) combination of the kinematic and stochastic models (lines 11 and 23). The mathematical reasoning regarding the calculation of distances, direction, location and lane from the vehicles can be obtained in subsections 4.2.1.1 e 4.2.1.3.

Before presenting each of these novelties in detail, we consider it necessary to determine when there is a latent risk situation between the Ev and Ov vehicles. Therefore, taking into account the kinematic model presented in Algorithm 2 (lines 7,19), a situation is considered a risk when the following conditions are present:

- the safety distance (d_{sf}) is greater than the longitudinal distance (d_a) between a pair of Ev and Ov vehicles;
- the Ev and Ov vehicles move in a straight line; and
- the lateral distance (d_p) between the Ev and Ov vehicles is less than the maximum lateral distance constant (l_d).

The formal definition of each variable can be found in subsection 4.2.1.1.

In relation to the first specific novelty, the inclusion of vehicle's steering angle information in the kinematic model allows to improve the conditions that define a risk situation. In order to establish whether or not a driver should make a lane change, in this model it is necessary to determine if all vehicles (including Ev) move in a straight line. Therefore, a low value in the steering angle ensures that the vehicles remain moving in a straight line. This condition in the kinematic model is determined by $St_{Ev} < St_k$ and $St_{Ov} < St_k$, where St_k is a constant to be determined experimentally. The case $St_{Ev} > St_k$ assumes that Ev is moving in a curve.

Regarding the second specific novelty, the new condition to determine the direction of the vehicles in the kinematic model considers the possibility that an Ov vehicle around Ev may have another direction with respect to the two established in Algorithm 1 (same and opposite directions). This new condition seeks to determine a *caution alarm* when Ev and Ov are moving closer in a curve or when an Ov near Ev makes a lane change. In this new condition, the safety distance in a d_{sc} curve (line 28 in Algorithm 2) is defined as:

$$r_{ack} = \frac{L}{\tan(St_{Ev})} \quad (5.3)$$

$$d_{sc} = d_{critical} + r_{ack} \quad (5.4)$$

In Equations 5.3 and 5.4, r_{ack} represents the Ackerman steering model's radius, where L is the longitudinal distance between the vehicle's tires and St_{Ev} is the steering angle of the Ev . Consequently, d_{sc} is the result of summing r_{ack} with $d_{critical}$, which is the value of the product

between the velocity of Ev (V_{Ev}) and the reaction time of a driver t_r . The distance d_{EOv} (line 31 in Algorithm 2) corresponds to the Euclidean distance between the Ev and Ov vehicles.

In relation to the third specific novelty, the combination of the two kinematic and stochastic models is, in our view, the most representative contribution of this thesis. As mentioned in section 5.3, the combination of these models ensures a better system response. In this sense, the context necessary for the definition of the *PGM* and its combination with the kinematic model is explained below.

5.4.2 Probabilistic graphical model

In this contribution, the *PGM* elected is a *DBN* represented by a *2-time slice* structure ($2 - TBN$). The $2 - TBN$ structure allows representing a dynamic transition of the stochastic model in a single period of time, which reduces the complexity of the *PGM*. Therefore, in the construction of the $2 - TBN$ structure, the considerations presented in subsections 5.1.1 and 5.2.1 are applied. Consequently, the first step is to define the hypothesis variable and the random variables.

In this proposal, the hypothesis variable is defined based on the main objective of the assistance system, that is, to avoid collisions. In this case, the assistance system must prevent Ev from colliding with other Ovs around it on a straight line. When there is a latent risk situation (i.e., lines 7 and 19 in Algorithm 2), the system must inform the Ev driver (i.e., lines 12 and 24 in Algorithm 2). This information is the product of the inference of a *state* of the hypothesis variable. The discrete hypothesis variable is defined as *Decision* (D). This Decision variable is made up of the following states: *i*) warning (w); *ii*) change (ch); and *iii*) same (s). Figure 20 shows each of these states. The inference of each of these states in the system model will depend on the random variables.

The definition of the discrete random variables depends on the information of the kinetic model that can produce one of the states of the hypothetical variable *Decisions*. The random discrete variables defined for the $2-TBN$ model are: *i*) Lane (L); *ii*) Vehicle direction (Vd); and *iii*) Location (Lc). A variable *Lane* has three states (*both*, *other* and *same*); the variable *Vehicle direction* also has three states (*caution*, *danger* and *alert*); However, the *Location* variable has only two states (*multiplex localization* and *simple localization*). The definition of each of these states is made from information obtained from the kinematic model presented in subsection 5.4.1.

In order to establish each state of the random variables, three steps are necessary: *i*) obtain and calculate the constants of the kinematic model necessary to establish the relation between Ev and a single Ov (direction, lane and localization); *ii*) storing the values of the constants obtained for a given number of Ov vehicles; and *iii*) determine the states of the random variables of the *PGM* (L , Vd and Lc).

Figure 20 – Decision's states

Source: Elaborated by the author.

5.4.2.1 Obtaining and calculating parameters

To obtain and calculate the parameters, the first step is to know the kinematic information of E_v and O_v s. This information is obtained through in-vehicle sensors and the messages between vehicles through V2V communications. The second step is to determine the values of the parameters (lines 2-8 and 15-20) *direction*, *lane* and *localization* for each O_v during an [Algorithm 2](#) execution routine. Each of these parameters can only store the following values: *i*) direction (same or opposite directions); *ii*) lane (same or other lane); and *iii*) localization (front, back and side).

To obtain the value of the *direction* parameter, two conditions (lines 2 and 15) are established in [Algorithm 2](#). The *line 2* determines the *same direction* value, while *line 15* determines the *opposite direction* value. The formal explanation of these two conditions can be obtained in subsections [4.2.1.1](#) and [4.2.1.3](#).

To calculate the value of the *lane* parameter, the first step is to determine, by means of the kinematic model (lines 6 and 18), what is *Ov*s lane with respect to *Ev*. At this point, the kinematic model establishes a relation between the lateral distance (d_p) and lateral safety distance (d_{ls}) defined by Equations 4.10 and 4.11 presented in subsection 4.2.1.1. This relationship establishes the following:

$$lane \quad Ov = \begin{cases} d_p < d_{ls} & , same \quad lane \\ d_p > d_{ls} & , other \quad lane \end{cases} \quad (5.5)$$

Finally, to determine the value of the *localization* parameter it is considered the algorithm presented in subsection 4.2.1.2. In lines 8 and 20 from Algorithm 2, the localization algorithm response is generalized to determine only the location of an *Ov* with respect to *Ev* on the *front*, *back* and *side*.

5.4.2.2 Storage of parameter values obtained

After determining the values of the three parameters, the next step is to obtain the states of the random variables. This process consists in storing the parameter values (*direction*, *lane*, *localization*) for each *Ov* present around *Ev*. In this procedure, the environment around *Ev* is scanned. We assume, in this thesis, that the context of greater risk for *Ev* is a highway with only two lanes (i.e., this limitation is due to the simulation scenario in which the system was evaluated), where *Ev* is surrounded by *Ovs*. This reference context allows us to establish a maximum number of five *Ovs* around *Ev*. Figure 20a is a near example of this situation. All values of the parameters obtained during the five readings of the environment are stored in vectors through Algorithm 3.

Algorithm 3 presents the *critical class* implemented in the system. Its main functions are to diagnose the period of obtaining each reading of the environment, store the value of the parameters obtained and determine the states of each random variables of the *PGM*. In the *LANECARS* method it is necessary for the diagnostic procedure to obtain the values of the parameters during a consecutive time. If this is not the case, this procedure resets the values of the vectors and the *scan constant*, which takes into account the number of readings of the environment. The diagnostic procedure ensures that the environment around *Ev* is scanned.

Afterward, the *LANECARS* method compares the values obtained and acts in two conditions: *i*) if the values obtained are not from the same vehicle, it stores the values of the parameters in their corresponding vector; and *ii*) if the parameter values correspond to the same vehicle, the *LANECARS* method updates the values of the parameters in the corresponding vector. In lines 9 and 21 from Algorithm 2, it is executed the *class critical* represented in Algorithm 3.

Algorithm 3 – Critical

```

1: function LANE_CARS(scan, constans, vectors, t, id)
2:   diagnostic t
3:   scan = scan + 1
4:   compare new data with vectors
5:   save the new data in vectors
6:   return scan, vectors
7: end function
8:
9: function LANES(vector – lane)
10:  Determine the state of variable Lane.
11:  return lane's state ( $L_{st}$ )
12: end function
13:
14: function DIRECTION(vector – direction)
15:  Determine the state of variable Vehicle direction
16:  return direction's state ( $Vd_{st}$ )
17: end function
18:
19: function LOCALIZATION(vector – localization)
20:  Determine the state of variable Localization
21:  return localization's state ( $Lc_{st}$ )
22: end function

```

5.4.2.3 Determination of the random variables states

Finally, through the *Lanes*, *Directions* and *Localization* methods of the *critical class*, the states of the random variables are determined. In each of these methods, the vectors of each parameter (*lane*, *direction* and *localization*), which store the values pre-set in subsection 5.4.2.1, are used. In these methods, it is calculated how many times a specific value is present in the vector of each parameter. With this information, it is possible to determine each state of each random variable, defined in the penultimate paragraph of subsection 5.4.2. Charts 1, 2 and 3 present the determination of the random variables states.

Chart 1 – Random variable Lane (*L*)

<i>Random Variable Lane</i>		
constant lane		<i>Lane's states</i>
Value same	Value other	
False	False	<i>NA</i>
False	True	<i>other</i>
True	False	<i>same</i>
True	True	<i>both</i>

Source: Research data.

In each of the Charts (1, 2 and 3), a *True* value corresponds to *at least one specific value*

Chart 2 – Random variable Vehicle-direction (Vd)

Random Variable Vehicle-direction		
constant direction		Vehicle-direction's states
Value same direction	Value opposite direction	
False	False	<i>NA</i>
False	True	<i>alert</i>
True	False	<i>caution</i>
True	True	<i>danger</i>

Source: Research data.

Chart 3 – Random variable Location (Lc)

Random Variable Location			
constant localization			Location's states
Value front	Value back	Value side	
False	False	False	<i>NA</i>
False	False	True	<i>simple localization</i>
False	True	False	<i>simple localization</i>
False	True	True	<i>multiplex localization</i>
True	False	False	<i>simple localization</i>
True	False	True	<i>multiplex localization</i>
True	True	False	<i>multiplex localization</i>
True	True	True	<i>multiplex localization</i>

Source: Research data.

that is present in the vector of a parameter. Consequently, the *False value* corresponds to *no presence of a specific value* in the vector of a parameter. Finally, the relation of the variable hypothesis with the random variables is established again through the possible values of the parameters *lane* and *direction*. The justification, to determine the relationship between variables was considered from the proposal submitted by Schubert (2012).

5.4.2.4 Learning the 2-TBN model structure

Given the complexity of learning the structure of a graphical model, as already stated in subsections 5.1.1 and 5.2.1 to define the structure of the 2 – TBN model, the following steps were performed: *i*) creation of a dataset; *ii*) adaptation of the dataset created for learning the structure of the 2 – TBN model; and *iii*) learning of the structure of the 2 – TBN model using the methodology *score-based learning*.

In the first step, for the creation of the dataset, five collision situations were developed using a new version of the simulator presented in subsection 4.2.2. The five collision situations used were:

- lead vehicle moving at lower constant speed;

Chart 4 – hypothesis variable Decision (*D*)

<i>hypothesis variable Decision</i>				
constant lane		constant direction		<i>Decision's states</i>
Value same	Value other	Value same direction	Value opposite direction	
False	True	False	True	<i>same</i>
False	True	True	False	<i>same</i>
False	True	True	True	<i>same</i>
True	False	False	True	<i>change</i>
True	False	True	False	<i>change</i>
True	False	True	True	<i>change</i>
True	True	False	True	<i>warning</i>
True	True	True	False	<i>warning</i>
True	True	True	True	<i>warning</i>

Source: Research data.

- following vehicle making a lane change maneuver;
- vehicles making (and not making) a lane change maneuver with opposite direction;
- vehicles on two-way road; and
- following vehicle making a lane change maneuver with traffic on the two lanes.

A video showing each of the collision situations can be found in https://www.youtube.com/watch?v=P_ASXVc7eGI.

In each of these situations, during four hours, fifteen *Ov* vehicles interact with *Ev* performing the collision maneuver. Each collision situation is simulated independently. During the simulations of collision situations, the random variables determined in subsection 5.4.2.3 are recorded through the implementation of the kinematic model presented in subsection 5.4.1. The records obtained by each simulation were joined in a single file (dataset.csv) at the end of the simulations. Then, in this final file, the states of the random variables *Lane* and *Vehicle direction* are used to obtain the states of the hypothesis variable (*Decision*). For this automatic procedure, we used the python data analyze library *pandas* (AUGSPURGER *et al.*, 2017). In total, the dataset has 25.332 states records for each of the variables (i.e., *Decision*, *Lane*, *Vehicle direction*, *Location*).

In the second step, the dataset obtained in the previous stage must be adapted to be used with the 2 – *TBN* model. Considering Brownlee (2017b), it is possible to restructure a time series dataset as a supervised learning problem by using the value at the previous time step to predict the value at the next time-step. Furthermore, Brownlee (2017a), shows how it is possible to do this using python. In this contribution, it was considered the sliding window method proposed by Brownlee (2017a), with a window width equal to one, taking into account the structure of the 2 – *TBN* model (two time steps or slices). The datasets created in the

previous two steps (*dataset* and *DBN – k2*) can be found in <https://github.com/AndresEGH/Cooperative-driving-assistance-system-for-lane-change.git>.

Finally, in the third step, to define the structure of the 2 – *TBN* model, the *score-based learning* methodology was used to create the structure. The justification for this decision is because the constraint-based structure learning methodology is quite sensitive to noise in the data, which can give rise to the mistaken independence assumptions and cause the network to miss out on finding the structure (KARKERA, 2014). Therefore, we consider that the methodology score-based learning is the best option. The algorithm chosen from this methodology to create the structure of the 2 – *TBN* model was *K2*.

Considering Sucar (2015), the *K2* algorithm incrementally adds parents to each node, as long as it increases the global score. When adding parents to any node does not increase the score, the search stops. Also, given a causal ordering it guarantees that there are no cycles in the graph. The choice of this algorithm is determined by two important motivations: *i*) the ability of the software tool used, which in our case was proposed by Gonzales, Torti and Wuillemin (2017); and *ii*) the result obtained with the algorithm *K2*. Figure 21 presents the models of the structures obtained using the algorithm *K2*.

(a) *BN* model structure using the *K2* algorithm and dataset from step one.

(b) 2 – *TBN* model structure using the *K2* algorithm and DBN-K2 dataset from step two.

Figure 21 – 2 – *TBN* model structure

Source: Elaborated by the author.

Figure 21 shows how the structure of the model in Figure 21a is present into the structure of each *time-slide* of the model shown in Figure 21b. In the 2 – *TBN* model shown in Figure 21b, the *time-slide t-1* is composed of the variables $D_{-1}, L_{-1}, Vd_{-1}, Lc_{-1}$, while the *time-slide t* is composed of the variables D, L, Vd, Lc . In Figure 21b it can also be observed that there are only edges of the variables of *slide t-1* toward the variables of *slide t*. Therefore, the structure of the 2 – *TBN* model proposed in this contribution has the two main assumptions of a *DBN* defined in section 5.2.

5.4.2.5 Learning of the 2-TBN parameters

The learning of the parameters for the 2 – TBN model is made from the adapted dataset (*DBN – k2.csv*) and the structure of the 2 – TBN model obtained in the previous subsection. In order to obtain these model parameters, we used the maximum likelihood estimation approach defined in subsection 5.1.1. The choice of this approach is mainly due to the compatibility of the library proposed by Ankan and Panda (2015) with the driving simulation platform used in this thesis. The parameters (*CPDs*) obtained for each of the variables, in the proposed 2 – TBN model, can be found in <<https://github.com/AndresEGH/Cooperative-driving-assistance-system-for-lane-change/blob/master/CPD's.txt>>.

5.4.2.6 Inference in the 2-TBN model

Finally, at this point, it is possible to estimate the effect on the Decision (*D*) hypothesis variable, considering evidence acquired in the random variables [Lane (*L*), Vehicle direction (*Vd*), Location (*Lc*)] in the model 2 – TBN. In this contribution, the exact inference algorithm Junction tree was used, through MAP queries (subsection 5.1.2) in order to obtain one of the states of the Decision hypothesis variable. Considering Bellot (2016), the main advantage of this algorithm is its computational efficiency, since this algorithm transforms the *PGM* into a tree with a set of properties, which allow obtaining the estimates (probabilistic or *MAP* queries). In this contribution, *the inference class* made corresponds to the prediction of the future state of the *Decision* variable (i.e., variable *D* in the *time – slide t*), from the current evidence obtained for the variables *Lane*, *Vehicledirection* and *Location* (i.e., variables *L_1*, *Vd_1* and *Lc_1* in the *time – slide t – 1*). Algorithm 4 shows *the Inference class* implemented in the system.

Algorithm 4 – Inference

```

1: function DBN( $L_{st}, Vd_{st}, Lc_{st}$ )
2:   Structure 2 – TBN
3:   CPD's for model's variable
4:   MAP query
5:   return Decision's state ( $D_{st}$ )
6: end function

```

Its main function is to estimate the state of the *Decision* variable that will generate information for the driver in Algorithm 2 (lines 12 and 24). In *DBN* method, current evidence obtained from the random variables (L_{st}, Vd_{st}, Lc_{st}) are received. In addition, this method incorporates the 2 – TBN model structure and its parameters. Finally, the *DBN* method also contains the code needed to perform the *MAP* query. This code is based on the library proposed by Ankan and Panda (2015).

5.4.3 System design

The previous version of the platform (presented in subsection 4.2.2) did not present sufficient characteristics that allowed the implementation of the system proposed in this thesis. Therefore, for the development of the cooperative driver assistance system for lane change was necessary to create new characteristics in the driving simulation platform. These new features in the structure are listed below:

Chart 5 – Comparison between the two versions of the driving simulation platform

Previous version	New version
The main script can configure: Vehicle with its sensors. Vehicles' working interface. An initial position of the vehicle. An initial vehicle orientation.	In this version of the main script are configured, beyond the parameters of the old version, the following vehicle initial items: Velocity. Lane. Waypoint. Course direction.
In the Self-driving algorithm, a proportional control was implemented. In the self-driving algorithm, vehicles can only change lanes in one direction (right or left).	In the self-driving algorithm, a PID control was implemented. The self-driving algorithm allows vehicles to change lanes in both directions.
Vehicles may be disabled in the simulation by a collision or obstruction of their course.	To avoid vehicle disabling, in this version a reset algorithm was implemented for all vehicles.
The user can not interact solely by using the keyboard (without commands) with the vehicles.	In this version was implemented an algorithm to control some actions of the vehicles only with the keyboard.
Creation of the NIC sensor.	Creation of two new sensors and their respective publisher for ROS.
Communication V2V without restrictions in the Time delay.	V2V communication with restrictions in the time delay.

Source: Research data.

Again, the simulations were executed in a laptop Dell Vostro 5470, with processor Intel Core i7 2.0 GHz and 64 bit, memory of 8GB, video card NVIDIA Geforce with 2 GB, and operating system Ubuntu 14.04 Trusty Tahr.

In this new version, a platform user can adjust more parameters of the initial behavior of the vehicles. This technical improvement favored the usability of the driving simulation platform, mainly, in the work with multiple vehicles in the simulations. In the previous version, the configuration of the parameters of velocity, lane and waypoint was done directly in the vehicles' 3D model in Blender. Therefore, the initialization of the simulations was slower. The course direction parameter is a novelty on this version. With this parameter, it was possible to

define situations with an opposite direction of the vehicles' movement.

The self-driving algorithm has the objective to execute the driver's activities, within the scenario, during the data collection (see subsection 4.2.3). In this new version of self-driving algorithm, The Proportional Integral Derivative (PID) controller implemented applies an accurate and responsive correction to a control steering angle of each vehicle's model. The PID controller is defined in Equation 5.6 as:

$$S_t = -K_p \cdot S_{tc} - K_i \cdot \int S_{tc} dt - K_d \cdot \frac{dS_{tc}}{dt} \quad (5.6)$$

Where, S_t is the control variable steering angle; S_{tc} is the current error value of steering angle; and K_p , K_i , K_d are the coefficients for the proportional, integral, and derivative terms, respectively.

The implementation of reset and keyboard control algorithms in each vehicle was aimed at improving the performance of the experiments. On the one hand, we have the reset algorithm that initializes the original conditions of a vehicle (position, velocity, lane, course direction and waypoint) each time a vehicle does not continue its course in the simulation. This is usually due to a collision between vehicles. On the other hand, we have the keyboard control algorithm that acts on each of the vehicles in order to have access its camera, change the lane, increasing or lowering the velocity of the vehicle and manually restarting each vehicle if necessary.

The creation of the two new sensors is due to the need to obtain the steering angle and current lane parameters of a vehicle. These sensors have the advantage of directly publishing the values obtained in ROS. The steering angle parameter is directly used in the kinematic model described in subsection 5.4.1. The lane parameter is used in the following subsection for the validation of the results.

Finally, in the V2V communication model, we assume a fifth of the time delay value obtained dynamically in the model presented in subsection 4.2.2.1. This decision is due to the results found in subsection 4.2.3.2. To improve this aspect in vehicle communication, the communication's model must have more communication channels and a channel for its control (FESTAG, 2014). Figure 22 gives an overview of the new version of the driving simulation platform used in this contribution.

5.4.4 Simulation results

In this subsection, the tests and results obtained from applying the cooperative driver assistance system for lane change are described. In the experiments were taken into consideration the use of the system presented in the last subsection. Taking into account that the state of the art of the area emphasizes the importance of ensuring safe conditions for people in the tests (through resources and facilities suitable for their development), for the accomplishment of the

Figure 22 – New system design

Source: Elaborated by the author.

experiment proposed in this thesis, we used the simulation platform. In order to involve people in a real set of test, in addition to demanding high logistical and economic standards, it is crucial to ensure the safety of participants. Given the conditions of the space where this research was developed, the driver assistance system was not tested with people. In this section, in each test we consider the new version of the self-driving algorithm to emulate the driver using the E_v in the simulations. The specific difference between E_v and O_v s in the simulations is in the use of the proposed driver assistance model. Finally, in this subsection, the scenario used in subsection 4.2.3 is considered again.

Experiments

The cooperative assistance system for lane change was assessed using three of the five collision situations presented in subsection 5.4.2.4. These collision situations are: *i*) vehicles making (and not making) a lane change maneuver with opposite direction; *ii*) vehicles on two-way road; and *iii*) following vehicle making a lane change maneuver with traffic on the two lanes. In each of these collision situations a *normal* and *dense* traffic was considered during one hour of simulation. In all, fifteen O_v vehicles are used in addition to E_v . In each of these collision situations, E_v is the only vehicle that uses the driver assistance system.

In the simulations of the three coalition situations, each O_v moves autonomously throughout the scenario during the execution time. The initial velocity and lane conditions in the O_v s vary during the simulation. This is because the scenario can change the speed and lane of vehicles

by using *checkpoints*. These characteristics were implemented in order to approximate the simulations with real situations. The scenario used in the simulations does not present climatological changes. Finally, it is important to note that all vehicles share a Basic safety message (BSM) among them through V2V communication, which contains information about vehicle id, position, speed, orientation, vehicle proportions and the current vehicle's lane. This latter information is used to validate the classification of the information presented to the driver in a *matrix of confusion*.

Results

In the experiments carried out applying the model of driver assistance system proposed we considered the evaluation of performance of the classification of information presented to the driver during E_v 's route on a highway. This evaluation helped to determine if the information presented to the driver corresponds to the real situations that can occur on highways. This correspondence is quantified through a *confusion matrix*. The *Confusion matrix on line calculator* proposed by Vanetti (2007) was used in the analysis of results.

In the simulation scenario, there is only one track with two lanes (*outside* and *inside* lanes). Each lane in the scenario is defined by a set of *waypoints*, which are distributed over each virtual lane of the track (Figure 23). In our simulations, all vehicles use the self-driving algorithm, and therefore they all move through a lane of the scenario. To know if a highway lane, around E_v , is occupied or not by an O_v , the O_v s share the information of its current lane using the *BSM*. Consequently, when the assistance system generates information for the driver, the information of the tracks of the other vehicles is compared with the information of the current lane of E_v . All this process is done for the purpose of evaluating the information provided by the driver assistance system. The current E_v lane information is obtained through a *Lane* sensor.

In addition, with the aim to compare the results of the classifications obtained with the proposed 2 – *TBN* model, another *PGM* was also implemented; created from the Bayesian network shown in Figure 21a. The main idea is to observe the differences between the two probabilistic graphical models, while they are used and evaluated under the same conditions. The parameters of the Bayesian network structure used can be found in <https://github.com/AndresEGH/Cooperative-driving-assistance-system-for-lane-change/blob/master/CPD's_BN.txt>.

Figures 24, 25 and 26 show the confusion matrix for each collision situation, in a normal and dense traffic, with respect to the two *PGMs* used (*BN* and 2 – *TBN* models). The information needed to create each confusion matrix can be found in <https://github.com/AndresEGH/Cooperative-driving-assistance-system-for-lane-change/tree/Datasets/Collision_situations>. The classes in each of the figures correspond to the information for the driver, including: i) **Class 1**: *warning*; ii) **Class 2**: *change*; and iii) **Class 3**: *same*. In general terms, in each of these figures, it is possible to observe an *accuracy* measure greater than 90% for each of the models tested. Moreover, in these figures is also possible to observe the measurement of agreement between

Figure 23 – Definition of lanes in the simulation scenario.

Source: Elaborated by the author.

classes through a *Kappa coefficient*. A *kappa coefficient* always has a value less than or equal to one. At this point it is possible to observe a *Kappa coefficient* value greater than 0.8 in all Figures. According to Altman (1991), a value between 0.8 and 1.0 for the *Kappa coefficient* is a significant agreement between the observed values.

In order to look for more specific indicators in the performance measurement of each of the *PGMs* tested, the *Precision*¹ and *Recall*² parameters are also considered. In addition, in order to compare the performance of the two models evaluated in each collision situation, the parameter *F1 score*³ is used. Tables 5, 6 and 7 present the values of this parameter, which is obtained from the confusion matrices of Figures 24, 25 and 26.

In the first collision situation (Figure 24) it is possible to observe that the *Precision* and *Recall* parameters in all the confusion matrices show a performance above 80%, for each of the models, in the respective traffic levels. This can be motivated by the considerable difficulty of the atypical situation that is evaluated. In relation to the *F1 score*, Table 5 shows how the relationship between *Precision* and *Recall* does not have significant changes for each evaluated model, with respect to the two levels of traffic considered. Finally, the *F1 score* values also allow us to verify that the performance of the two models (*BN* and 2 – *TBN*) is close, even with the slight difference in favor of the *BN* model.

¹ Precision it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (*PPV*) (BROWNEE, 2014).

² Recall it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate (*TPR*) (BROWNEE, 2014).

³ The *F1 score* is the harmonic mean, or weighted average, of the precision and recall scores (HACKELING, 2014).

Figure 24 – Collision situation 1: vehicles making (and not making) a lane change maneuver with opposite direction. **class 1**: warning; **class 2**: change; **class 3**: same.

Source: Elaborated by the author.

Figure 25 – Collision situation 2: vehicles on two-way road. **class 1**: warning; **class 2**: change; **class 3**: same.

Source: Elaborated by the author.

Figure 26 – Collision situation 3: following vehicle making a lane change maneuver with traffic on the two lanes. **class 1**: warning; **class 2**: change; **class 3**: same.

Source: Elaborated by the author.

Table 5 – *F1 score* for the collision situation 1

Class	BN model		2-TBN model	
	Normal traffic	Dense traffic	Normal traffic	Dense traffic
warning	0.97	0.96	0.91	0.91
change	0.88	0.90	0.87	0.86
same	0.93	0.94	0.93	0.91

Source: Research data.

Table 6 – *F1 score* for the collision situation 2

Class	BN model		2-TBN model	
	Normal traffic	Dense traffic	Normal traffic	Dense traffic
warning	1.00	0.97	1.00	0.99
change	0.97	0.96	0.92	0.93
same	0.99	0.97	0.99	0.97

Source: Research data.

Table 7 – *F1 score* for the collision situation 3

Class	BN model		2-TBN model	
	Normal traffic	Dense traffic	Normal traffic	Dense traffic
warning	0	0.96	0	0.94
change	1	0.98	1	0.97
same	1	0.97	1	0.95

Source: Research data.

In the second collision situation (Figure 25) it is possible to verify that the measure of the *Precision* parameter, for the *change* class in the 2 – TBN model, has the lowest value (86.36%) in a normal traffic. In the other classes, for the two models in the two traffic levels, the value of the *Precision* and *Recall* parameters is above 90%. In Table 6, again, there are no significant changes in the relation of the *Precision* and *Recall* parameters in each of the models with respect to the level of traffic. In this collision situation it is even more evident that the performance of the two models evaluated is practically the same.

Finally, in the third collision situation (Figure 26) in the confusion matrices used for the two models, with a normal traffic level (Figures 26a and 26b), the experiments did not register corresponding values for the warning class. This is due to the development of the experiment of this collision situation, that is, all vehicles in the road scenario are distributed throughout the runway of the simulation on both lanes. This obtained value indicates that during the time (one hour) of development of the experiment, the necessary conditions to present an information to the driver of the type *warning* did not come forward. However, as it can be seen in each of the confusion matrices, each of the remaining classes has a measure in the

Precision and *Recall* parameters above 90%, as in the second collision situation. In relation to the parameter *F1 score*, in Table 7 can be verified one more time the two characteristics mentioned in the first and second collision situations. Therefore, at this point is possible to conclude that the two models, besides having a good performance, do not present considerable differences. A video with some scenes from the experiments developed can be found in <https://www.youtube.com/watch?v=TZN-SIaPNhg>.

5.5 Final considerations

In this chapter we have presented a brief description of the theory of the graphical probabilistic models used in this thesis. In addition, we also presented the combination of the kinematic model with a graphical probabilistic model. We can consider as two contributions for thesis the following aspects: the choice of the graphic probabilistic model 2 – *TBN* for the cooperative system of assistance to the driver and the evaluation of the effect of this model in relation to another non-dynamic graphical probabilistic model.

The choice of the probabilistic model 2 – *TBN* was due to the possibility of representing, in a dynamic way, the transition of the variables of a Bayesian network between two consecutive time periods. With this theoretical definition of the 2 – *TBN* model, which is not known to have been previously used in the area, we have established another procedure to model the uncertainty implicit in the lane change problem. In this chapter, the procedure considered for the development of the 2 – *TBN* model can be found in detail.

In addition, to determine the effect of the 2 – *TBN* model, we established a direct comparison with a normal Bayesian network model, which also performed the same task of classifying information presented to the driver, under the same conditions. The results obtained from this comparison showed significant performance in both models in the classification of the information. Even though we did not obtain a decisive difference in the evaluation between the two probabilistic graphical models established, the comparison performed in this chapter made it possible to highlight the relevance of performing this type of procedure to obtain greater reliability of the results. Finally, it should be emphasized that the graphical probabilistic models, with respect to other outstanding models (WANG; YEUNG, 2016), have semantics and transparent properties, which allow a flexible approach to the uncertainty of a real problem.

OTHER CONTRIBUTIONS

In this chapter, other academic contributions developed during the doctoral period are presented. These contributions were made in collaborative work with LRM members¹. The first contribution was presented in the IEEE Intelligent Vehicles Symposium Proceedings 2014 ([GOMEZ *et al.*, 2014](#)) and the second was submitted this year as a short paper to IEEE Transactions on Intelligent Vehicles.

6.1 Traffic lights detection and state estimation using hidden markov models

To extract the traffic lights and their states from the image, the disturbance in the image and the uncertainty must be controlled in the best possible way. In this context, we proposed the image processing and the estimation of encountered states, applying the *Viterbi algorithm* ([RABINER, 1989](#)) in order to find the best sequence of states that exists with the traffic light detected. The obtained results, after applying the proposed method, have allowed to increasing the stability in the determination of current traffic light state, especially in video image sequences which have a greater noise margin associated with the scene.

6.1.1 Method

Despite the limitations of image processing, it is still a suitable method for detecting objects, but requires other technique that improves its performance. This paper presents the *HMM*, as a complement of image processing, seeking to detect traffic lights and their states with a better performance, using the *Viterbi algorithm* to find the optimal state sequence associated

¹ Paulo V. Prado, Tiago C. dos Santos, Carlos Massera Filho, Daniela Alves Ridel, Francisco Alexandre Ribeiro de Alencar e Professor Fernando Santos Osório.

with the observation sequence obtained. The method proposed in (GOMEZ *et al.*, 2014) consists of six phases that are presented in Figure 27.

Each of these phases are: *i) image input*, the image is captured by the camera located on the Autonomuos Vehicle (AV); *ii) image processing*, this phase prepares the region of interest in the image for traffic light detection; *iii) traffic lights detection*, this phase determines the location of the traffic light, highlighting it within a rectangle; *iv) state detection*, this phase determines the state that is active in the traffic light location; *v) estimation of state obtained*, this phase establishes the parameters required to establish the next state in the system; and *vi) given state*, this phase presents the result of the obtained state.

Figure 27 – Flowchart of the proposed method.

Source: Elaborated by the author.

6.1.1.1 Image input and image processing

Figure 28 shows a frame of 640x480 pixels, taken by the camera located on the top of the AV. Each frame is converted from the RGB color space to gray scale, using Equation 6.1:

$$\gamma = (0.299 * R + 0.587 * G + 0.114 * B) / 100 \quad (6.1)$$

Where $R = \text{img}(i,j, \text{Color}[0])$, $G = \text{img}(i,j, \text{Color}[1])$, $B = \text{img}(i,j, \text{Color}[2])$ and $\text{img}(i,j, \text{Color}[k])$ is the captured image. Next, the result γ is converted to an unsigned integer data type of eight bits.

Considering the proposals applied in (JIE *et al.*, 2013; YU; BAI, 2012), it has been considered 2/5 of the image height, with the purpose of removing some objects within the image that can interfere with the traffic light states (i.e., the lamps of vehicles). Next, a *median filter* is applied on the image resulted from the previous step, where for each pixel (i, j) it has been calculated the median $m(i, j)$ of values in a 3×3 neighborhood of (i, j) . After, assigned $I_m(i, j) =$

Figure 28 – Image input.

Source: Elaborated by the author.

$m(i, j)$, where I_m is the filtered image.

It is also necessary to eliminate the presence of the sky, some buildings and poles within the region of interest. Therefore, a *morphological gradient* (I_{mrg}) was applied, such as it is described in Equations 6.2 and 6.3 below:

$$kernel = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (6.2)$$

$$I_{mrg}(i, j) = I_{dil}(I_m, kernel) - I_{ero}(I_m, kernel) \quad (6.3)$$

where,

$$I_{dil}(i, j) = \max_{(\hat{i}, \hat{j}): kernel(\hat{i}, \hat{j}) \neq 0} I_m(i + \hat{i}, j + \hat{j}) \quad (6.4)$$

$$I_{ero}(i, j) = \min_{(\hat{i}, \hat{j}): kernel(\hat{i}, \hat{j}) \neq 0} I_m(i + \hat{i}, j + \hat{j}) \quad (6.5)$$

$I_{dil}(i, j)$ and $I_{ero}(i, j)$, determining the morphological operations of dilation and erosion. The next step involves the use of *threshold*, which helps to filter some pixels on the image that are not required. The threshold applied is an integration (through an openCV function) between the Otsu's algorithm (OTSU, 1979) and a *threshold binary* defined in Equation 6.6:

$$I_{th}(i, j) = \begin{cases} 255 & \text{if } I_{mrg}(i, j) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.6)$$

The $I_{th}(i, j)$ represents the resulting image after applying the threshold. Finally, seeking to improve the contours of the objects that are within the region of interest, *operation of erosion* (described in Equation 6.5) is applied using the following kernel:

$$kernel = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad (6.7)$$

This kernel was selected because its performance provided the best results in the implementation of erosion operation . Figure 29 is an example of processing applied to each of the frames.

Figure 29 – Image processing.

Source: Elaborated by the author.

6.1.1.2 Traffic lights detection

After the system processed the image, traffic lights were detected through their own geometric features. The first step that is done is a topological analysis of the image processing result, by a border following proposed in (SUZUKI *et al.*, 1985). Later, using Green's theorem (used on a plane through an open CV function), it is necessary to find the area of each figure found in the region of interest with the topological analysis; that establishes:

$$\oint_C (Pdx + Qdy) = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dxdy \quad (6.8)$$

Where, Q and P are functions of (x, y) defined on an open region.

Results obtained from each area in the region of interest establish two thresholds to discard figures whose area is not found in the range stated to be a traffic light. The two thresholds established ($th1$ and $th2$, where $th1 < th2$) were determined by the characteristics found in the

acquired images for different traffic lights distances in the region of interest. With elimination of a large proportion of figures with this condition, we considered the rectangular shape of the traffic lights. To obtain the geometric characteristics of rectangle, *the height* (H) and *the width* (W) were searched, calculating the dimensions by means the coordinates of the edge of each figure starting from the upper right corner, obtained from the topological analysis conducted. Then, a ratio H/W is calculated.

With the values of H and W , the area of elements is calculated once again in order to calculate a ratio of the area obtained above, and the area calculated with the parameters $H*W$. Finally, it is established a new condition in which the value of the relationship H/W must be between $th3$ and $th4$ (where $th3 < th4$) and the ratio of the areas calculated must be above a threshold ($th5$). This threshold was obtained by calculations on the images belonging to the traffic lights. If the above three conditions achieved, the element of the region of interest is a traffic light and the selected figure is highlighted by a rectangle around it. In Figure 30 it is possible to appreciate the traffic light detection of the image given in the Figure 28.

Figure 30 – Traffic light detection.

Source: Elaborated by the author.

6.1.1.3 State detection

Once the region is selected by the system, it is converted to HSV color space. To segment light bulbs with different colors it is applied a threshold to the hue component only. Cheng *et al.* (2001) uses the HSV color that is especially efficient when the images have non-uniform illumination such as shade (i.e, bad weather conditions), since hue is independent on intensity values. Furthermore, color segmentation in the 1-D hue space is computationally faster than in the 3-D RGB space. The HSV color space is arranged in a scale from 0 to 360. In the implementation used by *OpenCV* this range decreases to values between 0 and 180. The intervals ranging from 170 to 180 together with the interval between 0 and 20 define the color red. The yellow color is defined by the interval between 20 and 40. Finally the green color is set to the range between 41

and 90. To determine the state of traffic light, the system performs a pixel counting. The color with the largest number of pixels found defines the state. The minimum amount of pixels was set to 12. We defined this number after analyzing 35 samples with different sizes and statistically calculated the smallest number of pixels for the definition of a valid color.

6.1.1.4 Estimation of state obtained and given state

Here is where we use the Hidden Markov Model (HMM) proposed to improve the results obtained. This method is used primarily because it is desired to maintain the viability and flexibility of image processing, and it is here where the *HMM* proved to be an efficient mathematical tool which helps to avoid the problems of image processing. The first step was to create a Markov chain, which describes the phenomenon of the changes of the traffic light state. For this, a Markov chain of 4 states was proposed and can be seen in [Figure 31](#).

Figure 31 – A Markov chain with 4 states.

Source: Elaborated by the author.

In [Figure 31](#), it is also possible to see the state *N.D. (Not Detection)*. In the detection of traffic light and their states, the state *N.D.* is implicit, either because it does not actually present a traffic light and therefore their states, or because some limitation in processing the images does not allow for a possible state. The next step is to determine each element of the *HMM* ([RABINER, 1989](#)):

- N , is the number of states in the model. The individual states are denoted as $S = \{S_1, S_2, \dots, S_N\}$;
- M , is the number of distinct observation symbols per state;
- the state transition probability distribution $A = \{a_{ij}\}$;
- the observation symbol probability distribution in state j , $B = \{b_j(k)\}$; and

- the initial state distribution $\pi = \{\pi_i\}$.

In this contribution, were taken into account the following elements for the *HMM*:

$$\text{states} = \{\text{Red, Yellow, Green, N.D.}\}$$

$$\pi_i = \{0.25, 0.25, 0.25, 0.25\}$$

Table 8 – State transition probabilities.

States	Red	Yellow	Green	N.D.
Red	0.5	0	0.3	0.2
Yellow	0.3	0.5	0	0.2
Green	0	0.3	0.5	0.2
N.D.	0.25	0.25	0.25	0.25

Source: Research data.

Table 9 – The observation probabilities.

States	Red	Yellow	Green	N.D.
Red	0.8	0	0	0.2
Yellow	0	0	0	0
Green	0	0	0.8	0.2
N.D.	0.2	0.2	0.2	0.4

Source: Research data.

For determining each parameter, it has been taken into consideration the probability of the initial states is the same for each state. Also, it has been determined a higher probability for a transition and observation of the same state. Other relevant information is to consider that the probability of observation of the Yellow state is not representative for the system. Finally, it has been considered the second basic problem of the *HMM*, in order to obtain a more coherent state, starting from a state's sequence provided by the image processing and using the technique of *Viterbi's algorithm* (RABINER, 1989). The complete procedure can now be stated as following:

Initialization

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (6.9)$$

Where O_1 , it is the first element of the given observation sequence.

$$\psi_1(i) = 0 \quad (6.10)$$

Where $\psi_1(i)$, it is an array.

Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (6.11)$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (6.12)$$

Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (6.13)$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \quad (6.14)$$

Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1. \quad (6.15)$$

The resultant state in the final step (given state in [Figure 27](#).) of the system is more consistent, because the resultant state considers the results of image processing of five previous frames. Consequently, the system is less vulnerable to limitations inherent of the image processing.

To evaluate the performance of traffic light detection system, several urban sequences acquired with a camera on the vehicle were used, making a total of 649 images. The images were divided in five groups, one for each traffic light path, respectively. The images for testing were acquired with a *PointGrey Bumblebee 2* camera, with lenses 3.8 mm, 65° opening and a resolution of 648x488 pixels at 25 frames per second. Processing was executed in an industrial computer with processor intel Core i7, 8Gb RAM, mounted on the top of the CaRINA 2 ([FERNANDES et al., 2014](#)). To evaluate the method, the sequences of images were manually labeled to generate a ground truth.

Based on the images taken of one traffic light, it is possible to observe how the Viterbi algorithm use improves the detection of the traffic light state. In [Tables 10 and 11](#) the results of using the method proposed in this study are presented. The HMM introduction allows a general increase of 5.85% in the detection result, as shown in [Table 11](#). When analyzing the improvement separately in each color it is possible to observe that the green detection rate was increased by 18.75%.

Table 10 – Confusion matrix of results without HMM

	Red	Green	N.D.
Red	86	0	1
Green	0	26	0
N.D.	1	6	0

Source: Research data.

Table 11 – Confusion matrix of results with HMM

	Red	Green	N.D.
Red	86	0	1
Green	0	32	0
N.D.	1	0	0

Source: Research data.

In conclusion, the use of the *Viterbi algorithm* helps to improve the detection of traffic light states, correcting the eventual errors due to the image processing use.

6.2 Influence evaluation of a driving assistance system on a driver

For this contribution, it was implemented the first case (*following*) of the study proposed by [Sebastian et al. \(2009\)](#). In the test drive, the *following* case occurs when two or more vehicles have the same orientation and direction on the same lane. [Algorithm 5](#) illustrates the situation.

Algorithm 5 – Following case

```

1: for each pair of vehicles ( $A$ ,  $B$ ) do
2:   if  $|\theta_A - \theta_B| < \beta$  then
3:     compute the distances  $d_p$ ,  $d_{ls}$  and  $d_a$ 
4:     find the follower  $f$  and the leader  $l$ 
5:     calculate safe distance  $d_{sf}$ 
6:     if  $d_a < d_{sf}$  then
7:       Collision warning
8:     end if
9:   end if
10: end for

```

6.2.1 Driving simulation platform configuration

[Figure 32](#) shows the configuration of the driving simulation platform's architecture. This configuration was the result of carefully planned experiments to validate the selected DAS model influence on a driver. These experiments (subsection [6.2.2](#)) required an interaction among

Figure 32 – Driving simulation platform configuration used to evaluate the *DAS*' influence in a driver.

Source: Elaborated by the author.

vehicles, a highway scenario, a new V2V communication network topology and the use of a joystick that would allow the interaction of a user with the driving simulation platform.

Inside of the *Robotic Simulator* rectangle, which represents the *MORSE* simulator in Figure 32, it is possible to observe the representation of four different vehicles. For the 3D model of the vehicles in *MORSE* it was considered the use of sensors (GPS, velocity, pose and *NIC*) as it is presented in subsection 4.1.2.

For this driving simulation platform configuration, in the message of the sensor Network Interface Communication (*NIC*), it was considered a new value. This new value is the distance that exists between the *Ev* and each vehicle present inside the highway scenario of the simulation. The distance is calculated in the *NIC* sensor using the vehicles' location data. This information is important because this value determines the communication range of the vehicles in the new communications model implemented in *NS – 3*.

The *Communication Network simulator* rectangle, that represents the *NS – 3* simulator in Figure 32, obtains the sent messages by the *NIC* sensor, again through *SOCKET-SS* (subsection 3.7). The communications model implemented in *NS – 3* represents, through nodes, each one of the vehicles used in *MORSE*. The structure of the V2V communication model allows the *Ev*'s node to receive messages from the *Ov*'s nodes present in the simulation scenario. If the distance value among the vehicles (obtained in the sent message by *NIC*) is greater than a range of 300m, the communications model determines that the *Ev*'s node cannot receive messages. When the *Ovs* enter the communication range of *Ev*, the communication begins. The V2V communications model do not allow the *Ev*'s node to send messages to the *Ov*'s nodes, as it can be observed in the rectangle *Communication Network simulator* in Figure 32.

When the *Ev*'s node receives a message, from an *Ov*'s node, the *Ev*'s node sends

this information for the *MIDDLEWARE* rectangle, through *SOCKET-MS*. In Figure 32, the *MIDDLEWARE* rectangle, which represents *ROS* is composed by the nodes: *i*) Network Simulator output; *ii*) *DAS*; and *iii*) Joystick. A *ROS* system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model. The *ROS*' nodes are different from the nodes used in *NS* – 3.

The node *Network Simulator output* receives a message obtained from *SOCKET-MS* and makes it available for the *ROS*' system. The *DAS* node is the piece of software in the *ROS*' system where it is implemented the model presented in subsection 6.2. For the model execution, the *DAS* node needs the data obtained from the *Network Simulator output* node and sensor data from *Ev*, which are obtained from *MORSE* (Figure 32). *MORSE* allows a direct communication with *ROS*. Finally, the *joystick* node allows the interaction between the user's and the driving simulation platform. In this node are implemented the configuration and the execution of the joystick functions needed in the experiments. As it can be observed in Figure 32, this node also communicates directly with *Ev* in *MORSE*.

6.2.2 Experiments

With the objective to validate how the implemented *DAS'* model influences a driver, two of thirty-seven potential collision situations were chosen (a lead vehicle stops and a vehicle changing lanes on the same direction). These collision situations are proposed by Harding *et al.* (2014a). Those collision situations are the result of several crash analysis based on the accidents data collected in the United States in 2009, which could have been avoided using a system with V2V communications technology. From this result, the risk typology in this section was developed. In subsection III-A presented in (HARDING *et al.*, 2014a) the other thirty-five collision situations can be observed.

For the evaluation of the *DAS'* model influence on a driver, postgraduate students from the *science institute of mathematics and computers (ICMC-USP)* were invited to participate in the experiment. It was not disclosed to the participants the main purpose of the experiment, and each participant was required to have a valid driver's license and driving experience of a manual gear shift. Only twenty postgraduate students voluntarily accepted the invitation to participate. From the 20 participants 7 were female and 13 were male, and the participants' age averaged 27.2 years (*sd* = 3.65 years). They were divided into two groups in order to verify the *DAS'* model influence in the two collision situations they were exposed to: *i*) without *DAS* (4 women, and 7 men); and *ii*) with *DAS* (3 women, 6 men), in order to verify the *DAS'* model influence in the two collision situations the drivers from both groups were exposed to.

For the implementation of the two collision situations, it was developed a virtual highway based on a specific segment of the highway *Washington Luiz* (without considering its real geometric characteristics), in the São Paulo state (from the bridge at the main access way to São Carlos Street until the U-turn at exit 240). In this virtual highway, the *GPS* coordinates of

the real highway segment were used in order to obtain the real shape of the highway structure. Moreover, *Blender's* tools were also used (see subsection 3.7) to add some characteristics in the virtual highway: *i*) asphalt under ideal conditions; *ii*) the grass around the highway; and *iii*) the use of objects and infrastructure present in the real highway. For the virtual highway scenario, it was considered a sunny environment without climatological changes and one-way lanes.

Four vehicles were used in the virtual highway scenario. Only one of these vehicles was driven by the participants (*Ev*) using the *joystick G27*, while the other three vehicles were moving autonomously. For the experiment, each participant received instructions on how to operate the *joystick G27*, which consisted of a steering wheel, accelerator, brake and clutch for the gear shift (See Figure 33). Before the experiment began and data were collected, the participants had time to adapt to the driving simulation platform.

Figure 33 – Hardware used in the influence evaluation of a Driver assistance system in a driver.

Source: Elaborated by the author.

The three vehicles that were moving autonomously were controlled by an implemented algorithm, which defined their paths within the virtual scenario. Two of these vehicles were used on the collision situations (a conventional red vehicle and a truck), and the third vehicle was used to distract the drivers (a conventional orange vehicle). The sequence of events of the simulation consisted of the first collision situation, the encounter of the drivers with the orange vehicle and the second collision situation. Next, it is made a detailed description of the characteristics considered for each collision situation:

6.2.2.1 The lead vehicle stopped

To configure the danger in this situation, it was developed a routine in the algorithm for the red vehicle moving autonomously. In this routine, the red vehicle detects the presence of *Ev* at a distance of 30m from its rear. It then activates the brakes in the vehicle controller used by *Blender* in the red vehicle model (see subsection 3.7.1), immediately decelerating the red vehicle until it is completely stopped.

When a participant starts the experiment, the *Ev* vehicle is parked on the virtual highway. After the participant drives for a while in the virtual scenario path, the user encounters the other three vehicles. Two of these vehicles, the red and orange, are moving in each lane of the highway (i.e., highway with just two lanes) at a velocity of 70km/h, which prevents an overtaking maneuver between them. This is the moment the red vehicle detects *Ev* and the first collision situation initiates.

In this collision situation, the participant from the group using the *DAS*' model receives a warning sound, which serves to alert the participant of the collision risk that the red vehicle ahead could represent. The sound signal is enabled when the condition in line six of [Algorithm 5](#) is accomplished. In [Figure 32](#) it is possible to observe the relationship between the node *DAS* in *ROS* with the sound signal (i.e., *AL*).

6.2.2.2 A vehicle changing lanes in the same direction

In this second collision situation, it was used a 3D model of a truck as one of the autonomous vehicles in the simulation. Here, the algorithm implemented for the truck first percepts (in a range of 15m) the presence of *Ev* approaching behind of it in a different lane. Then, the routine in the algorithm activates the steering control of the vehicle controller used by *Blender* in the truck model, causing the truck to move towards *Ev*'s lane. The sudden change of lane by the truck, moving at 70km/h on the highway, initiates the second collision situation of the experiment. Like in the first situation, a warning sound alerts the participants from the group using *DAS* of the truck's presence.

The orange vehicle's sole purpose is to distract the attention of the driver between the two collision situations, but without causing any further risks to the driver, who overtakes the orange vehicle without difficulties. A video of these experiments is available on https://www.youtube.com/watch?v=H4lXVlq_8II.

6.2.2.3 Discussion

To obtain an experience feedback, when the platform users completed the experiment, each participant completed a survey containing seven questions. With the survey results was possible to observe a worrying situation from two of the questions. In *Did you break the speed limit?* 65% of the users answered affirmatively. In *Did you make some dangerous maneuvers during the trial route?* 70% of the users also answered affirmatively. This result confirms that drivers between 21 and 30 years of age have more than 50% chance to cause an accident, as also seen in (BENGLER *et al.*, 2014), where it analyzed the driver assistance systems impact in the last thirty years.

[Figure 34](#) presents the results of two other questions from the survey. [Figure 34a](#) shows that the most dangerous situation for the driving simulation platform users was the first one (the lead vehicle stopped). In [Figure 34b](#) regarding the question *Did you think that the Driver*

(a) Did you crash with some vehicle?

(b) Did you think that the Driver Assistance System helped in the test

Figure 34 – Survey questions

Source: Elaborated by the author.

Assistance System helped in the test? only 56% of the users from the group with *DAS* considered the *DAS'* model useful in the test.

To evaluate the *DAS'* influence on a driver, the driving simulation platform made a *test log*, with the objective to obtain some data of each user during the experiment, described in subsection 6.2.2. The data obtained in *log* are *Ev's* velocity, *Ev's* brake application rate, and the initial time of the collision situation.

The value of the initial time of the collision situation helps to find the magnitude of *Ev's* velocity and *Ev's* brake application rate in the exact moment when beginning the collision situation inside of each user's *log* (see the vehicle situation and the truck situation in Figures 35 and 36). *Ev's* velocity and *Ev's* brake application rate are the only data considered to analyze the *DAS* influence in drivers.

In order to analyze the results, were considered the mean, median, standard deviation,

and the maximum and minimum values from the number of samples (i.e., participants) in the experiment by each group (second paragraph in subsection 6.2.2). These statistical parameters can be observed in Tables 12 and 13. In addition to these parameters, it was used a *bootstrap method* to determine a 95% confidence intervals ($Bmin$ and $Bmax$ in Tables 12 and 13), with the objective to compare the data set of Ev 's velocity and Ev 's brake application rate for each group. The *bootstrap method* is used for the results analysis, with small sample drawn from a distribution that differs from a normal distribution. In this approach, no assumptions were made about the original population (TRIOLA, 2011).

By comparing the statistical parameters' values shown in Tables 12 and 13 for Ev 's velocity data ($VCS1$ being Ev 's velocity in the first collision situation) it is possible to observe that the mean, the median, and the bootstrap's confidence intervals ($Bmax$ and $Bmin$) in the two groups do not have significant differences.

Moreover, for Ev 's brake application rate data ($BCS1$ being Ev 's brake application in the first collision situation) the values of the median, in Tables 12 and 13, show that the brake practically was not activated in both groups since the most probable value in the median was zero. This fact shows that, during the first collision situation, a considerable majority of the experiment participants, in both groups, did not react in the beginning of the collision situation.

Another aspect observed in Tables 12 and 13 for $BCS1$, was that the bootstrap's confidence intervals ($Bmax$ and $Bmin$) do not present a considerable difference. Therefore, these values can determine that the use of the DAS' model during the first collision situation did not have an impact on the driver. In Figure 34a it is possible to observe this fact, given that it was the red vehicle (used in the experiment to generate the first collision situation) that, according to the survey completed by the experiment participants, caused the majority of the crashes.

On the other hand, in the second collision situation, the differences between the statistical parameters' values in Tables 12 and 13 are considerable for Ev 's velocity data ($VCS2$, Ev 's velocity in the second collision situation). Only the standard deviation between the groups has a slight difference. The most considerable difference between the two groups can be found between the bootstrap's confidence intervals for $VCS2$.

Therefore, with a 95% of confidence, it is possible to evidence that in the beginning of the second collision situation the participants' group using the DAS' model was moving at approximately 20km/h slower than the other participants' group which did not use the DAS' model. Figures 35 and 36 represent this fact, considering the relation between the labels, the velocity (Ev 's velocity) and the truck situation.

Comparing the statistical parameters obtained for Ev 's brake application rate data ($BCS2$ being Ev 's brake application in the second collision situation) it was possible to verify that all the participants of the group that used the DAS' model, did not used Ev 's brake during the beginning of the second dangerous situation.

Table 12 – Results group without *DAS*

Parameter	VCS1 (Km/h)	BCS1(%)	VCS2 (Km/h)	BCS2(%)
Mean	68.98	11.32	95.07	0.35
Median	66.43	0.0	92.71	0.0
Std	17.02	24.49	28.86	1.18
Min	42.95	0.0	54.59	0.0
Max	101.72	80.0	165.33	3.94
BD min	60.28	1.75	81.98	0.0
BD max	79.61	32.74	116.41	1.07

Source: Research data.

Table 13 – Results group with *DAS*

Parameter	VCS1 (Km/h)	BCS1(%)	VCS2 (Km/h)	BCS2(%)
Mean	66.03	7.57	73.32	0.0
Median	62.11	0.0	77.91	0.0
Std	10.01	19.96	27.74	0.0
Min	57.44	0.0	34.57	0.0
Max	87.91	60.62	113.8	0.0
BD min	61.30	0.52	55.98	None
BD max	74.59	28.61	89.99	None

Source: Research data.

Figure 35 – A participant's log without DAS' model.

Source: Elaborated by the author.

Figure 36 – A participant's log with DAS' model.

Source: Elaborated by the author.

Consequently, after analyzing the experiment's results, it is possible to determine that the *DAS'* model, in the second collision situation, could have an influence on the driver. [Figure 36b](#) is an example of a participant's log from the group that used the *DAS'* model. In it is possible to observe that *Ev*'s brake was activated seconds after that the *DAS'* model adveted the driver.

Finally, considering that the two collision situations were consecutive, it could be determined that the experiments' participants who used the *DAS* model understood the model's effect only after the first collision situation.

6.3 Final considerations

In this chapter we have presented other two contributions during the doctoral period. These contributions helped define and develop the research presented in this thesis. In the first proposal of this chapter, it is possible to verify the improvement obtained in the detection of states of a semaphore by adopting the combination of a deterministic model and a probabilistic model. With regard to the second proposal also of this chapter, we present the first stage of development of the models of cooperative assistance to the driver, which was important to obtain the results in [chapters 4 and 5](#). In this proposal for cooperative assistance, we were faced with several difficulties in carrying out evaluations with a group of people (i.e., logistics, low participation and the attitude of people among others). From this experience we were able to identified how these types of difficulties can affect the times of the project and the validation of the necessary results.

CONCLUSIONS

According to the studies presented in this thesis, the development of Driver Assistance Systems has shown, in the last decades, the capacity to contribute to reducing the rate of deaths by traffic accidents in the world. A continuing challenge with these technologies is to make them move from research and developmental stages to the application stage on the roads. Currently, some government institutions, such as the *NHTSA*¹ in the United States, are promoting the use of these assistance systems in new vehicles offered to citizens. Consequently, it is possible to consider that the Driver Assistance Systems is a technological transition point between the vehicles operated by humans and the autonomous ones.

For the elaboration and evaluation of *DAS* proposed in this thesis was considered the development of our own driving simulation platform, based on studies in the area. The use of the driving simulation platform facilitates the work in the following aspects: safety, design time, economic, and logistical. Although the obtained results using a driving simulation platform cannot be considered definite, these tools could have best results if the models used for its components (e.g., sensors, vehicles, communications and so on) had feedback from obtained data in a real test field.

This feedback requires flexible driving simulation platforms (i.e., mainly in research) that allow the researcher to modify the characteristics in the setup of the driving simulation platform. Therefore, taking these question into account, our driving simulation platform allows its users to adjust its characteristics (e.g., hardware and software), for an experimental implementation, regarding technical and economic requirements. The applications presented in Sections 4.1, 4.2, 5.3 and 6.2 allow us to prove the flexibility of the driving simulation platform.

In relation to each of the systems covered in this thesis, in section 4.1 we evaluated the time delay effect on the communication with respect to the performance of the cooperative adaptive cruise control developed. In this contribution, we punctually verified how the increase

¹ <https://www.nhtsa.gov/equipment/safety-technologies>

in the time delay, in the communication, changes the performance of the proposed control. Therefore, in the model of the control system, it is possible to establish that the minimum time delay in the vehicular communication favors the performance of the CACC. This result coincides with another study in the area, where experiments similar to ours were carried out in a real field of test. In this sense, we can infer that results we obtained in a driving simulation platform are close to an experiment in the real field of test.

In [section 4.2](#), we presented a new cooperative approach for the collision warning among vehicles, with the objective to provide driver assistance. Using V2V communication again, we proposed a new method for identifying which is the leader and which is the follower between two vehicles that share kinematics information. Furthermore, a new algorithm was presented, in which the current location of one or some vehicles around an ego vehicle can be determine during its movement. The assistance model proposed in this contribution is totally deterministic.

Regarding assessment, this model was assessed through a new version of the driving simulation platform presented in [Chapter 3](#). In the communications model we implemented the time delay effect regarding the distance among the vehicles, which is present in a real V2V communication channel. To achieve this effect, we used experimental data from a study. Also, the communication model allowed a bidirectional communication among the E_v vehicle and the O_v vehicles. To ensure the location of the E_v vehicle in the simulation, a *Discrete Kalman Filter* was implemented to eliminate the noise added in the sensors, as to bring the simulation as close to reality as possible.

The results obtained in the simulation allowed us to see that the time delay effect, regarding the distance in a V2V communication channel, could affect the warnings in general, in the proposed system, when the E_v vehicle is in a vehicular congestion within the communication coverage. This effect has already been highlighted in state of the art. A possible solution to this simulation problem is the use of more channels in the communication model. Furthermore, the warning ranges obtained in the experiments confirmed that one of the most significant advantages of the V2V communication is a larger coverage. This coverage ranges in the vehicle detection could be applied in others driver assistance systems.

In [section 5.3](#), a new cooperative driver assistance system for the lane change was developed and evaluated. The main characteristic of this model is the fact that it combines a kinematic model with a stochastic one, represented by a graphical probabilistic model. The main function of the kinematic model is to estimate the risk situations that may occur around E_v through information acquired from the E_v sensors and basic safety messages sent by other vehicles through vehicular communications. Furthermore, the graphical probabilistic model has the main task of establishing what information will be presented to the driver, with respect to the risk situation established by the kinematic model. This kind of decision has to be stochastic given that the system depends directly on a human, whereby the system's response cannot be deterministic in nature.

The decision to consider a *PGM* in the development of the new *DAS* is mainly due to its compatibility with the problem proposed in this thesis. The *PGMs* is a technique of machine learning that allows structuring a model of flexible form from a dataset of variables considered by the developers. However, the greatest difficulty encountered by us in dealing with the *PGMs* was the lack of a detailed description of how to develop these models. In the state of the art, we often find studies that do not justify the development of *PGM* structures, which may generate questions about their results. Unfortunately, the books consulted on the subject do not consider real applications that complement the complexity of their explanations.

Considering the problem of not describing the models, we have described and applied the knowledge acquired in *PGMs* during the time of our research in [Chapter 5](#). As our main contribution to Driver Assistance Systems models, we have been able to structure a procedure to implement and evaluate the proposed *PGM 2-Time slice Bayesian Network (2 – TBN)*. Applying *2 – TBN* model, we can get a dynamic representation of a Bayesian model and make inferences without high time cost, which is a problem that happens with models of *DBNs* when are unrolled for $T > 1$. This aspect in *DBNs* is not feasible to solve a road safety problem.

In order to develop and evaluate our new model of driver assistance, in addition to developing a new version of our driving simulation platform, we consider it is important to compare our results with another *PGM*. Therefore, in the evaluation of our proposal, we also constructed a Bayesian network. These two stochastic models were used during simulated collision situations of the same type, with normal and dense traffic levels. The results obtained by applying the two models (*BN* and *2 – TBN*) allowed us to verify their positive performance during the execution of the experiments. The two models did not present significant differences, which allow us to infer that our proposed model is close to the performance of the model considered in the evaluation, therefore, it can be configured as another possibility of work for other researchers.

7.1 Future work

From the development of this thesis, as future works, we can highlight two objectives: to improve some aspects of the driving simulation platform used in this thesis, and also to improve the vehicles communication model.

Regarding the improvement of technical aspects of the driving simulation platform used in this thesis, we highlight: *i)* diverse weather conditions; *ii)* dynamic changes in the physical conditions of the roads; *iii)* development of events that incorporate pedestrians, motorcyclists and bicycle users; *iv)* improvements in the sensor models used; *v)* a new path finding algorithm for *Ovs*; and *vi)* realistic lighting effects.

In addition to these improvements, we also have the objective of creating the Vehicular Dynamic Model (*VDM*). The next version of the *VDM* in the driving simulation platform has

to consider some aspects, such as: *i)* to edit and easily replace, the different components of the *VDM* , *ii)* to use the main parameters of real vehicles, in the vehicles' model design, and finally, *iii)* the model must be developed to be stiff and efficient from a numerical point of view.

Another important aspect is to improve the vehicles communication model. In our model, we use a single communication channel, which is in charge of distributing the information contained in the basic safety message among all vehicles in the simulation, within the communications range. As a future work, we believe that the new model of *VANETs* must have multiple channels, among which there is a channel that controls the sending and receiving of information between the vehicles present in a road environment. In addition, it is also important to continue improving realism in communications by considering real phenomena such as, for example, the attenuation of the communication signal by the presence of physical barriers in the environment.

Finally, according to the results obtained in the developed *DAS* models, it was possible to determine that one of the future works will be considering the highway curvature, as another source of information within the kinematic models already established. This information will help improve the determination of the lane of an *Ov* around *E_v*.

BIBLIOGRAPHY

ABENDROTH, B.; BRUDER, R. Capabilities of humans for vehicle guidance. **Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort**, Springer, p. 3–18, 2016. Citation on page 83.

AEBERHARD, M.; RAUCH, S.; BAHRAM, M.; TANZMEISTER, G.; THOMAS, J.; PILAT, Y.; HOMM, F.; HUBER, W.; KAEMPCHEN, N. Experience, results and lessons learned from automated driving on germany's highways. **IEEE Intelligent transportation systems magazine**, IEEE, v. 7, n. 1, p. 42–57, 2015. Citation on page 46.

ALASMARY, W.; ZHUANG, W. The mobility impact in ieee 802.11p infrastructureless vehicular networks. In: **2010 IEEE 72nd Vehicular Technology Conference - Fall**. [S.l.: s.n.], 2010. p. 1–5. ISSN 1090-3038. Citation on page 53.

ALTMAN, D. G. **Practical statistics for medical research**. [S.l.]: Chapman and Hall/CRC press, 1991. Citation on page 101.

AMODIO, A.; PANZANI, G.; SAVARESI, S. M. Design of a lane change driver assistance system, with implementation and testing on motorbike. In: IEEE. **Intelligent Vehicles Symposium (IV)**, 2017 IEEE. [S.l.], 2017. p. 947–952. Citation on page 84.

ANDRADE, S. S. C. d. A. **Panorama dos acidentes de transporte terrestre no Brasil: das interações às sequelas e ao óbito: uma contribuição para a sua vigilância**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2015. Citation on page 30.

ANKAN, A.; PANDA, A. **Mastering Probabilistic Graphical Models Using Python**. [S.l.]: Packt Publishing Ltd, 2015. Citations on pages 80, 81 e 96.

AREM, B. V.; DRIEL, C. J. V.; VISSER, R. The impact of cooperative adaptive cruise control on traffic-flow characteristics. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 7, n. 4, p. 429–436, 2006. Citation on page 54.

AUGSPURGER, T.; BARTAK, C.; CLOUD, P.; HAYDEN, A.; HOYER, S.; MCKINNEY, W.; REBACK, J.; SHE, C.; HORIKOSHI, M.; VAN, J.; BOSSCHE, d. **Pandas**. 2017. Available: <<http://pandas.pydata.org/>>. Accessed: 24/10/2017. Citation on page 94.

BAI, F.; KRISHNAN, H. Reliability analysis of dsrc wireless communication for vehicle safety applications. In: **2006 IEEE Intelligent Transportation Systems Conference**. [S.l.: s.n.], 2006. p. 355–362. ISSN 2153-0009. Citation on page 62.

BAI, F.; STANCIL, D. D.; KRISHNAN, H. Toward understanding characteristics of dedicated short range communications (dsrc) from a perspective of vehicular network engineers. In: **Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking**. New York, NY, USA: ACM, 2010. (MobiCom '10), p. 329–340. ISBN 978-1-4503-0181-7. Available: <<http://doi.acm.org/10.1145/1859995.1860033>>. Citation on page 53.

BARBARESSO, J.; CORDAHI, G.; GARCIA, D.; HILL, C.; JENDZEJEC, A.; WRIGHT, K. **USDOT's Intelligent Transportation Systems (ITS) ITS Strategic Plan 2015-2019**. [S.l.], 2014. Citations on pages 32 e 62.

BARTELS, A.; MEINECKE, M.-M.; STEINMEYER, S. Lane change assistance. In: **Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort**. [S.l.]: Springer, 2016. p. 1235–1259. Citations on pages 84 e 85.

BEHERE, S.; TÖRNGREN, M.; CHEN, D.-J. A reference architecture for cooperative driving. **Journal of Systems Architecture**, Elsevier, v. 59, n. 10, p. 1095–1112, 2013. Citation on page 42.

BELLOT, D. **Learning Probabilistic Graphical Models in R**. [S.l.]: Packt Publishing Ltd, 2016. Citations on pages 80, 81 e 96.

BENGLER, K.; DIETMAYER, K.; FARBER, B.; MAURER, M.; STILLER, C.; WINNER, H. Three decades of driver assistance systems: Review and future perspectives. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 6, n. 4, p. 6–22, 2014. Citation on page 119.

BHARATI, S.; ZHUANG, W. **Link-Layer Cooperative Communication in Vehicular Networks**. [S.l.]: Springer International Publishing, 2017. 111 p. Citations on pages 53 e 54.

BIFULCO, G. N.; PARIOTA, L.; GALANTE, F.; FIORENTINO, A. Coupling instrumented vehicles and driving simulators: Opportunities from the drive in2 project. In: IEEE. **Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on**. [S.l.], 2012. p. 1815–1820. Citations on pages 46 e 47.

BLANA, E. A survey of driving research simulators around the world. Institute of Transport Studies, University of Leeds, 1996. Citations on pages 45 e 46.

BLENDER. **Blender 2.78 Manual, Vehicle Controller Physics**. 2016. Available: <https://www.blender.org/manual/game_engine/physics/types/vehicle.html>. Accessed: 05/09/2017. Citation on page 52.

BROWNLEE, J. **Classification Accuracy is Not Enough: More Performance Measures You Can Use**. 2014. Available: <<https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/>>. Accessed: 06/03/2018. Citation on page 101.

_____. **How to Convert a Time Series to a Supervised Learning Problem in Python**. 2017. Available: <<https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>>. Accessed: 24/10/2017. Citation on page 94.

_____. **Time Series Forecasting as Supervised Learning**. 2017. Available: <<https://machinelearningmastery.com/time-series-forecasting-supervised-learning/>>. Accessed: 24/10/2017. Citation on page 94.

BUTAKOV, V. A.; IOANNOU, P. Personalized driver/vehicle lane change models for adas. **IEEE Transactions on Vehicular Technology**, IEEE, v. 64, n. 10, p. 4422–4431, 2015. Citation on page 85.

CASTRO, C. **Human factors of visual and cognitive performance in driving**. [S.l.]: CRC Press, 2008. Citation on page 48.

CHALLEN, J. Reality bytes: Toyota's ultimate analysis tool. **Traffic Technology International**, 2008. Citation on page 46.

CHEN, Y.; ZHANG, G.; GE, Y. Formation control of vehicles using leader-following consensus. In: IEEE. **Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on**. [S.l.], 2013. p. 2071–2075. Citations on pages 37 e 56.

CHENG, H.-D.; JIANG, X. H.; SUN, Y.; WANG, J. Color image segmentation: advances and prospects. **Pattern recognition**, Elsevier, v. 34, n. 12, p. 2259–2281, 2001. Citation on page 111.

CHO, W.; HAN, K. S.; CHOI, H. K.; OH, H. S. Realization of anti-collision warning application using v2v communication. In: **2009 IEEE Vehicular Networking Conference (VNC)**. [S.l.: s.n.], 2009. p. 1–5. ISSN 2157-9857. Citation on page 61.

COUMANS, E. **Bullet 2.83 Physics SDK Manual**. 2015. Available: <https://github.com/bulletphysics/bullet3/blob/master/docs/Bullet_User_Manual.pdf>. Accessed: 05/09/2017. Citation on page 51.

DANG, R.; DING, J.; SU, B.; YAO, Q.; TIAN, Y.; LI, K. A lane change warning system based on v2v communication. In: IEEE. **Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on**. [S.l.], 2014. p. 1923–1928. Citation on page 84.

DOSHI, A.; TRIVEDI, M. M. Communicating driver intents: a layered architecture for cooperative active safety applications. In: IEEE. **Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on**. [S.l.], 2010. p. 373–378. Citation on page 36.

ECHEVERRIA, G.; LASSABE, N.; DEGROOTE, A.; LEMAIGNAN, S. Modular open robots simulation engine: Morse. In: IEEE. **Robotics and Automation (ICRA), 2011 IEEE International Conference on**. [S.l.], 2011. p. 46–51. Citations on pages 35 e 49.

ENGEL, T. **Use and validation of driving simulators**. Phd Thesis (PhD Thesis) — Fakultet for ingeniørvitenskap og teknologi, 2008. Citation on page 47.

ESPIÉ, S.; GAURIAT, P.; DURAZ, M. Driving simulators validation: The issue of transferability of results acquired on simulator. In: **Driving Simulation Conference North-America (DSC-NA 2005), Orlando, FL**. [S.l.: s.n.], 2005. Citation on page 47.

EU, E. C. **Cooperative, connected and automated mobility (C-ITS)**. 2016. Available: <https://ec.europa.eu/transport/themes/its/c-its_ga>. Accessed: 15/09/2017. Citations on pages 32 e 53.

EU, E. C. . D. G. for M.-T. **EU Road safety policy**. 2017. Available: <https://ec.europa.eu/transport/road_safety/specialist/policy_en>. Accessed: 12/11/2017. Citation on page 31.

_____. **Projects**. 2017. Available: <https://ec.europa.eu/transport/road_safety/specialist/projects_en>. Accessed: 12/11/2017. Citation on page 31.

_____. **Road fatalities in the EU since 2001**. 2017. Available: <https://ec.europa.eu/transport/road_safety/specialist/statistics_en>. Accessed: 10/11/2017. Citation on page 31.

FERNANDES, L. C.; SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; SALES, D.; MENDES, C.; PRADO, M.; KLASER, R.; MAGALHÃES, A. C.; HATA, A. *et al.* Carina intelligent robotic car: architectural design and applications. **Journal of Systems Architecture**, Elsevier, v. 60, n. 4, p. 372–392, 2014. Citation on page 114.

FESTAG, A. Cooperative intelligent transport systems standards in europe. **IEEE communications magazine**, IEEE, v. 52, n. 12, p. 166–172, 2014. Citation on page 98.

FISHER, D. L.; RIZZO, M.; CAIRD, J.; LEE, J. D. **Handbook of driving simulation for engineering, medicine, and psychology**. [S.l.]: CRC Press, 2011. Citation on page 45.

GIETELINK, O.; PLOEG, J.; SCHUTTER, B. D.; VERHAEGEN, M. Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations. **Vehicle System Dynamics**, Taylor & Francis, v. 44, n. 7, p. 569–590, 2006. Citations on pages 46 e 47.

GIMPEL, K.; RUDROY, D. **Statistical inference in graphical models**. [S.l.], 2008. Citations on pages 82 e 83.

GOMES, P.; OLAVERRI-MONREAL, C.; FERREIRA, M. Making vehicles transparent through v2v video streaming. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 2, p. 930–938, 2012. Citation on page 38.

GOMEZ, A. E.; ALENCAR, F. A.; PRADO, P. V.; OSORIO, F. S.; WOLF, D. F. Traffic lights detection and state estimation using hidden markov models. In: IEEE. **Intelligent vehicles symposium proceedings, 2014 IEEE**. [S.l.], 2014. p. 750–755. Citations on pages 107 e 108.

GÓMEZ, A. E.; GLASER, S.; ALAYLI, Y.; NETO, A. d. M.; WOLF, D. F. Cooperative collision warning for driving assistance. In: IEEE. **Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on**. [S.l.], 2016. p. 990–997. Citations on pages 49 e 54.

GÓMEZ, A. E.; SANTOS, T. C. dos; FILHO, C. M.; GOMES, D.; PERAFAN, J. C.; WOLF, D. F. Simulation platform for cooperative vehicle systems. In: **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2014. p. 1347–1352. ISSN 2153-0009. Citations on pages 49, 51 e 61.

GONCALVES, J. S.; ROSSETTI, R. J.; JACOB, J.; GONCALVES, J.; OLAVERRI-MONREAL, C.; COELHO, A.; RODRIGUES, R. Testing advanced driver assistance systems with a serious-game-based human factors analysis suite. In: IEEE. **Intelligent Vehicles Symposium Proceedings, 2014 IEEE**. [S.l.], 2014. p. 13–18. Citations on pages 46 e 47.

GONZALES, C.; TORTI, L.; WUILLEMIN, P.-H. aGrUM: a Graphical Universal Model framework. In: **International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems**. Arras, France: [s.n.], 2017. (Proceedings of the 30th International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems). Available: <<https://hal.archives-ouvertes.fr/hal-01509651>>. Citation on page 95.

GUILLET, A.; LENAIN, R.; THUILOT, B.; MARTINET, P. Adaptable robot formation control: adaptive and predictive formation control of autonomous vehicles. **IEEE Robotics & Automation Magazine**, IEEE, v. 21, n. 1, p. 28–39, 2014. Citation on page 37.

HACKELING, G. **Mastering Machine Learning with scikit-learn**. [S.l.]: Packt Publishing Ltd, 2014. Citation on page 101.

HARDING, J.; POWELL, G.; YOON, R.; FIKENTSCHER, J.; DOYLE, C.; SADE, D.; LUKUC, M.; SIMONS, J.; WANG, J. **Vehicle-to-vehicle communications: Readiness of V2V technology for application**. [S.l.], 2014. Citations on pages 71 e 117.

HARDING, J.; POWELL G., R.; YOON, R.; FIKENTSCHER, J.; DOYLE, C.; SADE, D.; LUKUC, M.; SIMONS, J.; WANG, J. **Vehicle-to-vehicle communications: Readiness of V2V technology for application. Report No. DOT HS 812014.** [S.l.], 2014. Citation on page 53.

HASSAN, B. **A design framework for developing a reconfigurable driving simulator.** Phd Thesis (PhD Thesis) — Paderborn, Universität Paderborn, Diss., 2014, 2014. Citations on pages 45, 46 e 48.

HORTON, G.; BIEDKA, M.; CHIFFI, C.; STASIO, C. de; DOLL, C.; MEIJA-DORANTES, L.; BRUHOVA-FOLTYNOVA, H.; SPERAT, Z.; JORDOVA, R.; BARTA, D. *et al.* Cooperative intelligent transport systems. 2016. Citation on page 53.

HUANG, C. M.; LIN, S. Y. An advanced vehicle collision warning algorithm over the dsrc communication environment: An advanced vehicle collision warning algorithm. In: **2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA).** [S.l.: s.n.], 2013. p. 696–702. ISSN 1550-445X. Citation on page 61.

IPEA, I. d. P. E. A.; PRF, P. R. F. Acidentes de trânsito nas rodovias federais brasileiras. caracterização, tendências e custos para a sociedade. IPEA, Brasil, 2015. Citation on page 30.

JEMAA, I. B.; GRUYER, D.; GLASER, S. Distributed simulation platform for cooperative adas testing and validation. In: IEEE. **Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on.** [S.l.], 2016. p. 77–82. Citation on page 39.

JERHOT, J.; FORM, T.; STANEK, G.; MEINECKE, M.-M.; NGUYEN, T.-N.; KNAUP, J. Integrated probabilistic approach to environmental perception with self-diagnosis capability for advanced driver assistance systems. In: IEEE. **Information Fusion, 2009. FUSION'09. 12th International Conference on.** [S.l.], 2009. p. 1347–1354. Citation on page 40.

JIE, Y.; XIAOMIN, C.; PENGFEI, G.; ZHONGLONG, X. A new traffic light detection and recognition algorithm for electronic travel aid. In: IEEE. **Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on.** [S.l.], 2013. p. 644–648. Citation on page 108.

KAREDAL, J.; CZINK, N.; PAIER, A.; TUFVESSON, F.; MOLISCH, A. F. Path loss modeling for vehicle-to-vehicle communications. **IEEE Transactions on Vehicular Technology**, v. 60, n. 1, p. 323–328, Jan 2011. ISSN 0018-9545. Citation on page 53.

KARKERA, K. R. **Building Probabilistic Graphical Models with Python.** [S.l.]: Packt Publishing Ltd, 2014. Citations on pages 80, 81 e 95.

KNAPP, A.; NEUMANN, M.; BROCKMANN, M.; WALZ, R.; WINKLE, T. Code of practice for the design and evaluation of adas. **Preventive and Active Safety Applications, eSafety for road and air transport, European Commission Integrated Project, Response**, v. 3, 2009. Citations on pages 46, 47 e 70.

KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. **Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.** [S.l.], 2004. v. 3, p. 2149–2154. Citation on page 35.

KOLLER, D.; FRIEDMAN, N. **Probabilistic graphical models: principles and techniques.** [S.l.]: MIT press, 2009. Citations on pages 77, 78, 81 e 82.

KUSANO, K. D.; GABLER, H. C. Safety benefits of forward collision warning, brake assist, and autonomous braking systems in rear-end collisions. **IEEE Transactions on Intelligent Transportation Systems**, v. 13, n. 4, p. 1546–1555, Dec 2012. ISSN 1524-9050. Citation on page 62.

LACAGE, M.; BU, J. **wave-simple-80211p.cc**. 2013. Available: <https://www.nsnam.org/doxygen/wave-simple-80211p_8cc_source.html>. Accessed: 06/09/2017. Citation on page 67.

LADEIRA, R. M.; MALTA, D. C.; NETO, O. L. d. M.; MONTENEGRO, M. d. M. S.; FILHO, A. M. S.; VASCONCELOS, C. H.; MOONEY, M.; NAGHAVI, M. Road traffic accidents: Global burden of disease study, brazil and federated units, 1990 and 2015. **Revista Brasileira de Epidemiologia**, SciELO Public Health, v. 20, p. 157–170, 2017. Citation on page 30.

LANG, B.; PARKES, A.; COTTER, S.; ROBBINS, R.; DIELS, C.; HIT, M. P.; KAPPLUSCH, J.; POSCHADEL, S. Train-all, benchmarking and classification of cbt tools for driver training. TRAIN-ALL consortium, 2007. Citation on page 46.

LAQUAI, F.; GUSMINI, C.; TONNIS, M.; RIGOLL, G.; KLINKER, G. A multi lane car following model for cooperative adas. In: IEEE. **Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on**. [S.l.], 2013. p. 1579–1586. Citation on page 37.

LEE, D.; HANSEN, A.; HEDRICK, J. K. Probabilistic inference of traffic participants' lane change intention for enhancing adaptive cruise control. In: IEEE. **Intelligent Vehicles Symposium (IV), 2017 IEEE**. [S.l.], 2017. p. 855–860. Citation on page 84.

LEGISLAÇÃO, C. dos D. **Código de trânsito brasileiro 4 edição, Capítulo III, Art. 60**. [S.l.]: Edições Câmara, 2010. Citation on page 30.

LU, N.; CHENG, N.; ZHANG, N.; SHEN, X.; MARK, J. W. Connected vehicles: Solutions and challenges. **IEEE Internet of Things Journal**, v. 1, n. 4, p. 289–299, Aug 2014. ISSN 2327-4662. Citation on page 53.

LUAN, T. H.; CAI, L. X.; CHEN, J.; SHEN, X. S.; BAI, F. Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks. **IEEE Transactions on Vehicular Technology**, v. 63, n. 3, p. 1419–1435, March 2014. ISSN 0018-9545. Citation on page 53.

LUO, Y.; XIANG, Y.; CAO, K.; LI, K. A dynamic automated lane change maneuver based on vehicle-to-vehicle communication. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 62, p. 87–102, 2016. Citation on page 85.

LYTRIVIS, P.; THOMAIDIS, G.; AMDITIS, A. Cooperative path prediction in vehicular environments. In: IEEE. **Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on**. [S.l.], 2008. p. 803–808. Citation on page 35.

MAAG, C.; MUHLBACHER, D.; MARK, C.; KRUGER, H.-P. Studying effects of advanced driver assistance systems (adas) on individual and group level using multi-driver simulation. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 4, n. 3, p. 45–54, 2012. Citations on pages 36, 46 e 47.

MARGARITIS, D. **Learning Bayesian network model structure from data**. Phd Thesis (PhD Thesis) — CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2003. Citation on page 80.

MCCALL, J. C.; WIPF, D. P.; TRIVEDI, M. M.; RAO, B. D. Lane change intent analysis using robust operators and sparse bayesian learning. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 8, n. 3, p. 431–440, 2007. Citation on page 39.

MECKLENBRAUKER, C. F.; MOLISCH, A. F.; KAREDAL, J.; TUFVESSON, F.; PAIER, A.; BERNADO, L.; ZEMEN, T.; KLEMP, O.; CZINK, N. Vehicular channel characterization and its implications for wireless system design and performance. **Proceedings of the IEEE**, v. 99, n. 7, p. 1189–1212, July 2011. ISSN 0018-9219. Citation on page 53.

MENHOUR, L.; D’ANDRÉA-NOVEL, B.; FLIESS, M.; GRUYER, D.; MOUNIER, H. An efficient model-free setting for longitudinal and lateral vehicle control: Validation through the interconnected pro-sivic/rtrmaps prototyping platform. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2017. Citation on page 38.

MIHAJLOVIC, V.; PETKOVIC, M. Dynamic bayesian networks: A state of the art. University of Twente, Centre for Telematics and Information Technology, 2001. Citation on page 82.

MURPHY, K. P. **Dynamic bayesian networks: representation, inference and learning**. Phd Thesis (PhD Thesis), 2002. Citations on pages 81 e 82.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012. Citations on pages 79 e 81.

NHTSA, N. H. T. S. A. **V2V communications fact sheet. Technical Report 11078-101414-v2a**. [S.l.], 2014. Citations on pages 32 e 62.

_____. **Vehicle to Vehicle Communications**. 2017. Available: <<https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communications>>. Accessed: 13/11/2017. Citation on page 32.

NIELSEN, T. D.; JENSEN, F. V. **Bayesian networks and decision graphs**. [S.l.]: Springer Science & Business Media, 2009. Citations on pages 77, 79, 80 e 81.

NILSSON, J.; FREDRIKSSON, J.; COELINGH, E. Rule-based highway maneuver intention recognition. In: IEEE. **Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on**. [S.l.], 2015. p. 950–955. Citation on page 85.

NILSSON, J.; SILVLIN, J.; BRANNSTROM, M.; COELINGH, E.; FREDRIKSSON, J. If, when, and how to perform lane change maneuvers on highways. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 8, n. 4, p. 68–78, 2016. Citations on pages 84 e 85.

NOTH, S.; EDELBRUNNER, J.; IOSSIFIDIS, I. An integrated architecture for the development and assessment of adas. In: IEEE. **Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on**. [S.l.], 2012. p. 347–354. Citations on pages 46 e 47.

OMAE, M.; FUKUDA, R.; OGITSU, T.; CHIANG, W.-P. Control procedures and exchanged information for cooperative adaptive cruise control of heavy-duty vehicles using broadcast inter-vehicle communication. **International journal of intelligent transportation systems research**, Springer, v. 12, n. 3, p. 84–97, 2014. Citations on pages 37 e 54.

ONERA; LAAS-CNRS; ISAE-SUPAERO. **What is MORSE?** 2017. Available: <https://www.openrobots.org/morse/doc/latest/what_is_morse.html>. Accessed: 14/09/2017. Citation on page 49.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 9, n. 1, p. 62–66, 1979. Citation on page 109.

PAN-WHO. **Report on Road Safety in the Region of the Americas**. [S.l.], 2015. Citation on page 29.

PAPP, Z.; BROWN, C.; BARTELS, C. World modeling for cooperative intelligent vehicles. In: IEEE. **Intelligent Vehicles Symposium, 2008 IEEE**. [S.l.], 2008. p. 1050–1055. Citation on page 40.

PASSCHIER, I.; NETTEN, B. D.; WEDEMEIJER, H.; MAAS, S. M.; LEEUWEN, C. V.; SCHACKMANN, P.-P. M. Ditem roadside facilities for cooperative systems testing and evaluation. In: IEEE. **Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on**. [S.l.], 2013. p. 936–942. Citations on pages 41 e 54.

PLOEG, J.; SCHEEPERS, B. T.; NUNEN, E. V.; WOUW, N. Van de; NIJMEIJER, H. Design and experimental evaluation of cooperative adaptive cruise control. In: IEEE. **Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on**. [S.l.], 2011. p. 260–265. Citations on pages 59 e 61.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE. **ICRA workshop on open source software**. [S.l.], 2009. v. 3, n. 3.2, p. 5. Citations on pages 50 e 68.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. **Proceedings of the IEEE**, Ieee, v. 77, n. 2, p. 257–286, 1989. Citations on pages 107, 112 e 113.

ROBERTS, K. M. Fhwa highway driving simulator. **Public Roads**, v. 44, n. HS-031 069, 1980. Citation on page 45.

ROCKWELL, T. H. Skills, judgment and information acquisition in driving. **Human factors in highway traffic safety research**, Wiley-Interscience, New York, p. 133–164, 1972. Citation on page 84.

RUDIN-BROWN, C. M.; WILLIAMSON, A.; LENNE, M. Can driving simulation be used to predict changes in real-world crash risk?[monash university accident research centre-report 299]. <http://www.monash.edu.au/miri/research/reports/muarc299.html> <http://www.monash.edu.au/miri/research/reports/muarc299.pdf>, Monash University Accident Research Centre, p. 1–40, 2009. Citations on pages 46 e 47.

RUSSELL, S.; NORVIG, P. **Artificial intelligence a modern approach**. [S.l.]: Prentice Hall, 2010. Citations on pages 78, 81 e 83.

SANTOS, F. P. **Identificação de sistemas neurais com redes bayesianas dinâmicas e transferência de entropia**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2017. Citation on page 82.

SANTOS, T. C. d.; GÓMEZ, A. E.; FILHO, C. M.; GOMES, D.; PERAFAN, J. C.; WOLF, D. F.; OSÓRIO, F.; ROSERO, L. A. A simulation framework for multi-vehicle communication. In: **2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)**. [S.l.: s.n.], 2015. p. 301–308. Citation on page 50.

SCHÖNER, H.-P.; MORYS, B. Dynamic driving simulators. **Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort**, Springer, p. 177–198, 2016. Citations on pages 44 e 45.

SCHUBERT, R. Evaluating the utility of driving: Toward automated decision making under uncertainty. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 1, p. 354–364, 2012. Citations on pages 40, 79 e 93.

SCUTARI, M.; DENIS, J.-B. **Bayesian networks: with examples in R**. [S.l.]: CRC press, 2014. Citation on page 81.

SEBASTIAN, A.; TANG, M.; FENG, Y.; LOOI, M. Multi-vehicles interaction graph model for cooperative collision warning system. In: **2009 IEEE Intelligent Vehicles Symposium**. [S.l.: s.n.], 2009. p. 929–934. ISSN 1931-0587. Citations on pages 61, 62, 64, 67, 86 e 115.

SINGH, S. **Critical reasons for crashes investigated in the national motor vehicle crash causation survey**. [S.l.], 2015. Citation on page 83.

SIVARAMAN, S.; TRIVEDI, M. M. Towards cooperative, predictive driver assistance. In: IEEE. **Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on**. [S.l.], 2013. p. 1719–1724. Citation on page 41.

_____. Dynamic probabilistic drivability maps for lane change and merge driver assistance. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 15, n. 5, p. 2063–2073, 2014. Citation on page 85.

SIVARAMAN, S.; TRIVEDI, M. M.; TIPPELHOFFER, M.; SHANNON, T. Merge recommendations for driver assistance: A cross-modal, cost-sensitive approach. In: IEEE. **Intelligent Vehicles Symposium (IV), 2013 IEEE**. [S.l.], 2013. p. 411–416. Citation on page 41.

SMITH, M.; WITT, G.; BAKOWSKI, D. A final report of safety vehicles using adaptive interface technology (task 15): Save-it summary and benefits estimation. **Kokomo, Indiana: Delphi Electronics and Safety**, 2008. Citations on pages 46 e 47.

STAVRINOS, D.; JONES, J. L.; GARNER, A. A.; GRIFFIN, R.; FRANKLIN, C. A.; BALL, D.; WELBURN, S. C.; BALL, K. K.; SISIOPIKU, V. P.; FINE, P. R. Impact of distracted driving on safety and traffic flow. **Accident Analysis & Prevention**, Elsevier, v. 61, p. 63–70, 2013. Citations on pages 46 e 47.

STILLER, C.; FARBER, G.; KAMMEL, S. Cooperative cognitive automobiles. In: IEEE. **Intelligent Vehicles Symposium, 2007 IEEE**. [S.l.], 2007. p. 215–220. Citation on page 39.

SUCAR, L. E. **Probabilistic Graphical Models: Principles and Applications**. [S.l.]: Springer, 2015. Citations on pages 79, 80, 81, 82, 83 e 95.

SUZUKI, S. *et al.* Topological structural analysis of digitized binary images by border following. **Computer vision, graphics, and image processing**, Elsevier, v. 30, n. 1, p. 32–46, 1985. Citation on page 110.

SWANSON, K. S.; BROWN, A. A.; BRENNAN, S. N.; LAJAMBE, C. M. Extending driving simulator capabilities toward hardware-in-the-loop testbeds and remote vehicle interfaces. In: IEEE. **Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE**. [S.l.], 2013. p. 115–120. Citation on page 35.

TEIXEIRA, F. A.; SILVA, V. F. e; LEONI, J. L.; MACEDO, D. F.; NOGUEIRA, J. M. Vehicular networks using the ieee 802.11 p standard: an experimental analysis. **Vehicular Communications**, Elsevier, v. 1, n. 2, p. 91–96, 2014. Citations on pages 68 e 73.

TRABELSI, G. **New structure learning algorithms and evaluation methods for large dynamic Bayesian networks**. Phd Thesis (PhD Thesis) — Université de Nantes; Ecole Nationale d'Ingénieurs de Sfax, 2013. Citation on page 80.

TRIOLA, M. F. **Elementary statistics technology update**. [S.l.]: Pearson Higher Ed, 2011. 360–388 p. Citation on page 121.

UTSUMI, A.; TADA, M.; YAMAMOTO, N.; MATSUO, N.; TORII, T.; SHIDOJI, K. Active safety with enhanced communication among traffic participants. In: IEEE. **Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on**. [S.l.], 2013. p. 1167–1172. Citation on page 42.

VANETTI, M. **Confusion matrix online calculator**. 2007. Available: <<http://www.marcovanetti.com/pages/cfmatrix/>>. Accessed: 06/11/2017. Citation on page 100.

VOLKSWAGEN. **Driver Assistance Systems**. 2017. Available: <<http://en.volkswagen.com/en/innovation-and-technology/technical-glossary/fahrerassistenzsysteme.html>>. Accessed: 13/11/2017. Citation on page 31.

WANG, H.; YEUNG, D. Y. Towards bayesian deep learning: A framework and some existing methods. **IEEE Transactions on Knowledge and Data Engineering**, v. 28, n. 12, p. 3395–3408, Dec 2016. ISSN 1041-4347. Citation on page 106.

WELCH, G.; BISHOP, G. An introduction to the kalman filter. department of computer science, university of north carolina. ed: Chapel Hill, NC, **unpublished manuscript**, 2006. Citation on page 68.

WESTHOFEN, D.; GRÜNDLER, C.; DOLL, K.; BRUNSMANN, U.; ZECHA, S. Transponder- and camera-based advanced driver assistance system. In: IEEE. **Intelligent Vehicles Symposium (IV), 2012 IEEE**. [S.l.], 2012. p. 293–298. Citation on page 40.

WHO, W. H. O. **Global status report on road safety 2015**. [S.l.]: World Health Organization, 2015. Citations on pages 29 e 30.

WYMAN, B.; ESPIÉ, E.; GUIONNEAU, C.; DIMITRAKAKIS, C.; COULOM, R.; SUMNER, A. Torcs, the open racing car simulator. **Software available at <http://torcs.sourceforge.net>**, 2014. Citation on page 48.

XIANG, X.; QIN, W.; XIANG, B. Research on a dsrc-based rear-end collision warning model. **IEEE Transactions on Intelligent Transportation Systems**, v. 15, n. 3, p. 1054–1065, June 2014. ISSN 1524-9050. Citation on page 61.

YAN, F.; WEBER, L.; LUEDTKE, A. Classifying driver's uncertainty about the distance gap at lane changing for developing trustworthy assistance systems. In: IEEE. **Intelligent Vehicles Symposium (IV), 2015 IEEE**. [S.l.], 2015. p. 1276–1281. Citation on page 86.

YOSHIMOTO, K.; SUETOMI, T. The history of research and development of driving simulators in japan. **Journal of mechanical systems for transportation and logistics**, The Japan Society of Mechanical Engineers, v. 1, n. 2, p. 159–169, 2008. Citations on pages 46 e 47.

YU, C.; BAI, Y. A traffic light detection method. In: **Advanced Technology in Teaching**. [S.l.]: Springer, 2012. p. 745–751. Citation on page [108](#).

ZARDOSHT, B.; BEAUCHEMIN, S.; BAUER, M. A. A decision making module for cooperative collision warning systems using vehicular ad-hoc networks. In: **16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)**. [S.l.: s.n.], 2013. p. 1743–1749. ISSN 2153-0009. Citations on pages [38](#) e [61](#).

ZEEB, E. Daimler's new full-scale, high-dynamic driving simulator—a technical overview. **Actes INRETS**, Institut national de recherche sur les transports et leur sécurité, p. 157–165, 2010. Citation on page [46](#).

ZHU, H.; SINGH, G. A communication protocol for a vehicle collision warning system. In: **Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCoM)**. [S.l.: s.n.], 2010. p. 636–644. Citation on page [61](#).