Master's Thesis

Conception, Implementation and Evaluation of the Safety-Critical Application „Intersection Movement Assist (IMA)" Based on Simulation and C-V2X Experimental Vehicles

Armando Miguel Garcia

Examiner: Prof. Dr. Hannah Bast
Adviser: Christian Massong, BA

Albert-Ludwigs-University Freiburg
Faculty of Engineering
Department of Computer Science
Chair for Algorithms and Data Structures

August 10th, 2020

Writing Period

05. 02. 2020 – 10. 08. 2020

Examiner

Prof. Dr. Hannah Bast – Albert-Ludwigs-Universität Freiburg

Second Examiner

Prof. Dr. Joschka Boedecker – Albert-Ludwigs-Universität Freiburg

Advisers

Christian Massong, BA– BMW AG

Declaration

I hereby declare that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.
I hereby also declare that my thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Munich, 10.08.2020

Place, Date

Signature

Abstract

Intersections make the flexible shaping and connection of our roads possible. Though accident statistics show that passing through an intersection is not a trivial task for vehicle drivers, as the majority of vehicle accidents happen inside or near to an intersection. This challenge becomes even greater, when the visibility of drivers is hindered due to weather conditions or obstacles, as for example buildings or other vehicles parked near to an intersection. One of the groups of road users that is especially threatened, even under normal road conditions, are the motorcycle riders. These often face the risk of being overlooked or misjudged by vehicle drivers.

The development of Advanced Driver Assistance Systems (ADAS) intends to solve this problem by extending the senses and capabilities of drivers, with the goal of reducing the probability of overlooking or misjudging other road users. One of the technologies to still be exploited in this field is Vehicle-to-Vehicle (V2V) communication. This technology relies on a wireless, non-line-of-sight dependent communication between vehicles, in contrast to existent ADAS solutions, which use conventional sensors for gathering information about their surroundings.

Of course, the application of such a technology for avoiding collisions at intersections is accompanied by several challenges and questions: Which properties should have such an application? Which one is the best approach for the prediction of collisions? How precise must be satellite navigation devices so that the application provides an acceptable prediction? Can we rely exclusively on the V2V-communication for issuing warnings to the drivers? Are intelligent traffic lights and map data needed for such an algorithm to work acceptably?

To answer these questions within this thesis, the whole process from drafting a V2V-version of the applications Intersection Movement Assist (IMA) and Left Turn Assist (LTA) until the validation using prototypes, takes place. This process includes the writing of an extensive concept for these applications, the development of four different approaches according to the most recent industry standards, and the implementation and comprehensive evaluation of those approaches in both a modern simulation environment and cutting-edge prototype vehicles. The obtained results demonstrate the great potential of V2V-communication for aiding vehicle drivers at intersections and thus preventing road accidents. Those also evince the strengths and weaknesses from each of the four implemented approaches, one of which demonstrated to be the more suitable for the conceived applications. It also becomes clear that map data is necessary for achieving acceptable levels of accuracy and that such an application should be seen as a sensor to be integrated into a larger system, instead of a standalone solution.

Contents

Terms and Definitions

Abbreviations

3GPP	3rd Generation Partnership Project
ABS	Anti-lock Braking System
ADAS	Advanced Driver Assistance System
API	Application Programming Interface
AT	Authorization Ticket
BSM	Basic Safety Message
CAM	Cooperative Awareness Message
CAN	Controller Area Network
C2C-CC	CAR 2 CAR Communication Consortium
C-ITS	Cooperative Intelligent Transport Systems
CMC	Connected Motorcycle Consortium
C-V2X	Cellular-V2X
DENM	Decentralized Environmental Notification Message
DGPS	Differential Global Positioning System
DoF	Degrees-of-Freedom
DRT	Driver Reaction Time
DSRC	Dedicated Short-Range Communication
DSS	Dynamic Safety Shield
DTC	Distance-to-Collision
DTI	Distance-to-Intersection
ETSI	European Telecommunications Standards Institute
GIDAS	German In-Depth Accident Study
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HMI	Human-Machine Interface
IMA	Intersection Movement Assist
IMU	Inertial Measurement Unit
IVI	In-Vehicle Information
LDM	Local Dynamic Map
LoS	Line-of-Sight

LTA	Left Turn Assist
LTE	Long Term Evolution
MAI	Motorcycle Approach Indication
MAP	Map Data
MAPEM	MAP Extended Message
MAW	Motorcycle Approach Warning
MDRT	Maximum Driver Reaction Time
MLT	Maximum Latency Time
MTTB	Mean Time-to-Break
OEM	Original Equipment Manufacturer
OSM	OpenStreetMap
PDU	Packet Data Unit
PET	Post Encroachment Time
PKI	Public Key Infrastructure
PoC	Point-of-Collision
PVS	Portable V2X-Setup
ROS	Robot Operating System
RSU	Road-Side Unit
SAE	Society of Automotive Engineers
SPaT	Signal Phase and Time
SPaTEM	Signal Phase and Time Extended Message
StVO	Straßenverkehrs-Ordnung (Road Traffic Regulations)
TPV	Time-Proximity Vector
TTC	Time-to-Collision
TTI	Time-to-Intersection
UE4	Unreal Engine 4
VRU	Vulnerable Road User
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

Symbols and Notations

Symbol	Unit	Meaning
a	m/s^2	Acceleration
d	m	Distance
t_{enc}	s	Encroachment time
ε	-	Error
g	m/s^2	Gravity of earth
h	$^\circ$	Heading
lat	$^\circ$	Latitude
lon	$^\circ$	Longitude
p	$\%$	Probability
t	s	Time
v	m/s	Speed
$\ll$	-	Much less-than
$(\cdot)_{ego}$	-	Value corresponding to the Ego-vehicle
$(\cdot)_{other}$	-	Value corresponding to the Other-vehicle

Definitions

Expression	Meaning
Blinker	Exterior light for indicating the intention of making a turn
Crash	Collision between two or more vehicles
Collision accident	Accident involving two or more road users
Cross-traffic	Traffic coming from the sides, i.e. right or left side
Driving accident	Single-vehicle accident
Ego-vehicle	Main vehicle regarding a specific scenario or layout
End-user	Operator of the vehicle in question
Emergency braking	Deceleration greater than 0.4 g or $3.9336\,\mathrm{m/s^2}$
Intersection	Portion of way, which is shared by two different roads
Latency	Delay between the occurrence of an event and its desired outcome
Motorcycle	Motor vehicle belonging to the European category L3e
OEM	Entity that builds, assembles, tests and validates the vehicle
Oncoming-traffic	Traffic driving in the opposite direction
Other-vehicle	Adversary of Ego-vehicle
Passenger car	Motor vehicle belonging to the European category M
Signalized intersection	Intersection controlled by a traffic light

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Motivation

The stagnating reduction of road accidents is a legitimate cause for concern. Alone in Germany, there were 30,174 accidents in 2018 involving motorcycles and resulting in personal injuries [1, 2]. This number has remained roughly constant since 2014 and hasn't improved as much as the number of car accidents in the last two decades.

Aiming for the improvement of road safety, many countries have joined the *Vision Zero*, a multi-national project with the goal of having zero fatalities and serious injuries caused due to road accidents before 2050. In order to achieve this goal, several measures are implemented, for example the minimization of the consequences of a traffic collision by improving the design and construction of motor vehicles, the modification to the road infrastructure for making crashes less probable and/or reducing their severity [3], the use of an Advanced Driver Assistance System (ADAS) for assisting vehicle drivers in the prevention of critical situations, and the general change on the behavior of road users for being more prudent and careful.

Motorcycles represent a special case with respect to passenger cars due to their scarcity of on-board sensors and their limited potential for minimizing the consequences of traffic collisions through mechanical design. This is one of the reasons why they are classified as Vulnerable Road Users (VRUs)[1], although they use exactly the same roads as passenger cars.

Figure 1.1 shows the 10 most common types of road accidents involving motorcycles. Those account for around 90 % of the total number of motorcycle accidents in Germany[2]. It becomes clear that the majority of collision accidents take place at intersections, motorcycle riders not being the main causer of this kind of accidents [2]. One of the reasons for this is that motorcycles are often overseen or misjudged by vehicle drivers. This makes the prevention of collision accidents on the motorcycle side very challenging and hard to address through a solution on the motorcycle side only. It also makes a point in favor of implementing a solution on the car side in order to prevent collisions with motorcycles.

[1] This also encompasses moped riders, cyclists, pedestrians and persons with disabilities or reduced mobility [4].
[2] This numbers are the result of the analysis of accidents registered in GIDAS in the period 2005-2017.

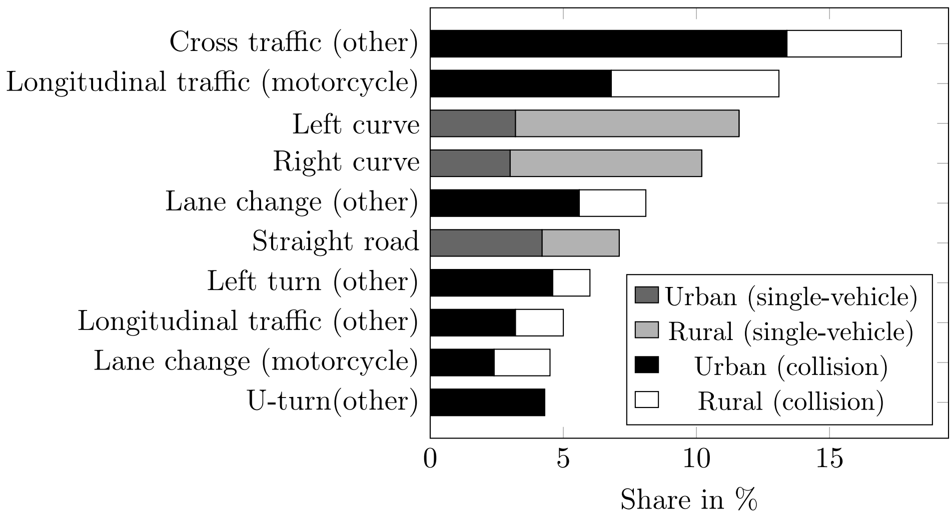


Figure 1.1: Road accidents involving motorcycles. Accidents are classified according to their place of occurrence. In parenthesis is the causer of the collision accident.

Solutions on the motorcycle side only have been proved effective when it comes to prevent single-vehicle motorcycle accidents [5], as only the motorcycle and its rider need to be taken into consideration. But problems arise when a motorcycle must not only consider the possible flaws of its rider, but also the mistakes of other road users, and that in complex environments like intersections, where conventional sensors (like cameras, radar and lidar) operate close to its perceptibility limits.

The Connected Motorcycle Consortium (CMC) is an international cooperation project aiming to increase motorcycle safety[3]. Being aware of the problem stated above, CMC acknowledges Vehicle-to-Vehicle communication as an effective way of improving motorcycle safety and also making motorcycles part of the future connected mobility.

1.2 State of the Art and Related Work

Being a relative new technology, V2V-communication for collision detection is still absent in the automotive market. At the time of this writing, there are only very few in-series produced vehicle models employing V2V-communication[4] [6, 7]. Although most Advanced Driver Assistance Systems for the prevention of accidents are based on conventional on-board sensor technologies, e.g. camera, radar or lidar, there are several companies, research institutes and organizations intending to exploit this technology in order to increase road safety.

[3] This project includes several motorcycle manufacturers, suppliers, industry associations and research institutes.
[4] V2V-communication is used on those vehicles for improving certain comfort and safety functions, although still not covering the detection of collisions of any kind.

Within the scope of the SimTD project[5], the evaluation of a prototypical cross-traffic-assistant solution based on V2V-communication is presented [8]. It is focused on the effects of the uncertainty of sensors and takes a full V2X-infrastructure for granted. Such an infrastructure does not only supports V2V but also Vehicle-to-Infrastructure (V2I)-communication, and foresees Map Data (MAP) and Signal Phase and Time (SPaT) messages being available to road users at every intersection. In the scope of the present work, a much practical approach is followed and the limitations and development constraints of such an advanced infrastructure are taken into consideration, which leads to the use of as less information as possible.

The project 5G Carmen is another effort for the implementation of V2X-applications. It is based on Vehicle-to-Network (V2N)-communication and aims to use the next generation of LTE-networks for the vehicles to share their information with the network. This information is then processed by a back-end server and individual messages are sent to the vehicles in order to inform about the need of a warning [9].

Regarding the detection of road collisions, two important measures have been established for assessing the criticality of a road situation: Time-to-Collision (TTC) and Post Encroachment Time (PET). The former is the time that a vehicle would need to collide with another vehicle if the current driving conditions are maintained. The latter is the time period from the moment a first vehicle leaves a conflict zone and until a second vehicle enters this same zone [10, 11].

The detection of collisions also finds wide use in the field of video games and three-dimensional modelling. The implemented approaches are classified in two main groups according to their calculation method: a-priori and a-posteriori [12]. A-priori means calculating the exact moment of collision by solving equations which represent the movement of the objects. A-posteriori means advancing one time unit at a time and updating the location of the objects in order to test if a collision happened.

1.3 Importance of a Reliable Algorithm for Collision Detection

Within the scope of V2X-communication, notifications and warnings can be given even before a threat or risk becomes clear and recognizable for the end-user. This makes the user-acceptance essential for the effectiveness of this kind of systems, as these applications cannot reach their maximum potential if drivers do not make usage of them in a sustainable and appropriate manner [13]. And because a frequent and voluntarily use of those technologies can only be achieved through a high level of acceptance, a reliable algorithm for issuing the notifications and warnings becomes a hard requirement.

[5] This was a cooperation project involving several manufacturers and suppliers of the automotive industry, which focused its efforts on the advancement of safe, intelligent and connected mobility.

This is also important for avoiding the so called "warning dilemma", which states that an ADAS issuing an unacceptable amount of false warnings would be considered as useless by the end-users and consequently deactivated.

Besides increasing the level of acceptance, a reliable warning is also important in order for the application not placing the driver in any danger, independent of the situation and context[6]. This becomes clear when we realize that a warning, which is either unnecessary or come at the wrong time, has the potential of distracting the user and may lead to an accident [14].

Due to production costs, the algorithm should be able to work while using a minimal amount of on-board sensors, as these can strongly drive the acquisition prices up. Avoiding adding too many on-board sensors is especially critical for the penetration into price-sensitive vehicle segments[7]. This highlights the need for the algorithm to be designed for reliability, as its output will not necessarily be contrasted with the readings of many on-board sensors.

1.4 Goal and Structure of the Thesis

The main goal of this thesis is the development and evaluation of an ADAS for the prevention of collisions at road intersections. This system uses direct vehicular-communication for issuing a warning to the end-user before a collision becomes unavoidable. The main topic of this thesis is a collision-avoidance algorithm which gathers information about the situation and the involved vehicles in order to estimate the probability of a collision. When suitable, this algorithm also issues a warning that is delivered to the end-user over a Human-Machine Interface (HMI).

For the greatest flexibility in terms of usage, this collision-avoidance algorithm must be as vehicle-dynamics-independent as possible, thus enabling its deployment on multiple vehicle classes without major changes. The algorithm should additionally be able to predict collisions by using a single received V2X-message. This is because of every V2X-capable vehicle having to hide their identity[8], which makes the association between multiple transmitted V2X-messages challenging [16].

In order to provide answers to the questions stated before and evaluate the potential of the different approaches proposed, the main components for the correct operation of this system are explained in chapter 2 (*Foundations*).

[6] A dangerous situation can only be avoided if the system does not distract the user (as less false positives as possible) and if it responds quickly enough when information is needed (very low levels of latency).
[7] Those segments, also including almost all motorcycles models, account for over two-thirds of the total vehicle population in Germany [15].
[8] In terms of C-ITS, vehicles must hide their identities by using pseudonyms when sending CAMs. Several pseudonym-change strategies are explained in [17]

The main work of this thesis is then presented in the three following chapters:

- Chapter 3 (*Conception*) explains the definition and goal of the application IMA, the covered use cases and the requirements to be fulfilled. It also presents four different approaches for implementing the collision-detection algorithm and explains the possible inclusion of the application LTA under the umbrella of IMA.

- Chapter 4 (*Implementation*) begins with the presentation of the main tools used for implementing and evaluating this application. It then goes on explaining the information and parameters needed for the algorithms to predict collisions and how the different approaches work.

- Chapter 5 (*Evaluation*) presents the results of the evaluation of every approach for collision detection, which are gathered by both using a simulation environment and conducting real driving tests. This chapter also explains the selection of the data-sets, reaction models and other parameters for the evaluation.

All chapters are then brought together and condensed in chapter 6 (*Summary*) by providing the most relevant insights gained during the writing of this thesis. This last chapter also states possible enhancements for the approaches and makes recommendations for future works.

2 Foundations

2.1 Communication

The communication between road users is the corner stone of this work, as it enables vehicles to disclose internal information about their current state. This information can be used by other entities (vehicles, RSUs, VRUs, etc.) for taking better decisions regarding their stance as part of a complex road system.

By using the received information, a road user can be informed about oncoming road hazards, even if he/she cannot yet perceive them as such (e.g. due to a limited field of view). This opens new possibilities for vehicular safety applications, because of the perceived environment not being limited by the Line-of-Sight (LoS) between road users and their surroundings, which is the case for existing solutions based on conventional sensors[1].

In order to enable a standardized implementation, the frequency range between 5.855 GHz and 5.925 GHz is reserved for this communication. This frequency range is divided into *control channels* for safety-related applications and *service channels* for comfort and information-related applications. The information used in this work for predicting and avoiding collisions is thus found in the *control channel*.

2.1.1 V2X-Communication

Vehicle-to-Everything (V2X)-communication refers to a vehicular communication system for the transference of information between a vehicle and another road entity. This term encompasses other more specific types of communications such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N) and Vehicle-to-Pedestrian (V2P).

At the moment of this writing, there are two main technologies for the implementation of V2X-communication: C-V2X and DSRC. And although they aim for the same goal and even use the same frequency spectrum for direct communication[2], they heavily differ from one another in the way they transmit and receive information over the physical layer.

[1] Vehicular communication, within the scope of Cooperative Intelligent Transport Systems (C-ITS), is not limited to the communication between road vehicles but also aims for the interaction between all road entities, this including vehicles, cyclists, pedestrians, traffic lights, traffic signs, trains, planes, and more [18].

[2] Both C-V2X and DSRC operate in the ITS 5.9 GHz spectrum band.

Cellular-V2X (C-V2X)

This technology is the one considered for this work, and also the one operating on the experimental vehicles used for the implementation and evaluation. It was developed by the 3rd Generation Partnership Project (3GPP), and is based on the usage of the cellular network for the communication between vehicles and their surroundings. It supports direct-communication through the so called PC5 interface, which enables the transmission of information without needing a connection to the cellular network [19].

The C-V2X standard is presently under further development. The latest releases are able to work on top of both LTE and 5G, the latter intended for strongly reducing latency times[3]. At the time of this writing, 3GPP releases 14, 15 and 16 are available. Release 14 being the first one covering direct communication, i.e. PC5 interface [19].

Dedicated Short-Range Communication (DSRC)

The second main technology for vehicular communication is DSRC. This is specified in the standard IEEE 802.11p, a modified version of the standard IEEE 802.11, better known as Wi-Fi. The MAC layer of DSRC is based on the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol and it operates by building an ad-hoc network for vehicles and road infrastructure to interact with each other [20].

2.1.2 Types of Messages

For using V2X-communication in the most efficient way and at the same time taking the complexity of road interactions into consideration, multiple kinds of messages are available for use. Every type of message serves a different purposes and works independent of the technology used for the transmission over the physical layer. Table 2.1 contains the most frequently used types of messages in the scope of V2X, CAM[4] being the most relevant for this work.

Message types like MAPEM and SPaTEM are not used by the here presented approaches because of their dependence on a well-developed road infrastructure[5]. IVI-messages, which belong to V2I-communication, are also not considered because of them being part of the road infrastructure and independent of the behavior of road users.

[3] The expected transmission latency over the cellular network lies under one millisecond.
[4] Both CAM and BSM are message types conceived for safety applications. CAM is defined by the European Telecommunications Standards Institute (ETSI) while BSM is defined by the Society of Automotive Engineers (SAE).
[5] The reliability of the solution would then be strongly linked to the presence of V2X-capable traffic lights and Road-Side Units (RSUs) at every intersection.

Within the scope of collision avoidance, DENMs[6] can also be used for the asynchronous transmission of awareness messages to other road users, when a collision is detected to be likely [21]. This type of messages is not considered as a reliable source for the detection of collisions because of their use for these purposes being optional [22].

Type	Definition	Specification
BSM	Basic Safety Message	SAE J2735_201603
CAM	Cooperative Awareness Message	ETSI EN 302 637-2
DENM	Decentralized Environmental Notification Message	ETSI EN 302 637-3
IVI	In-Vehicle Information	ISO/TS 19321:2015
MAPEM	MAP Extended Message	ISO/TS 19091:2017
SPaTEM	Signal Phase and Time Extended Message	ISO/TS 19091:2017

Table 2.1: Main types of messages for vehicular communication.

Cooperative Awareness Message (CAM)

This message type is conceived for enhancing the awareness of road users by facilitating the exchange of information about position, dynamics and other attributes among road entities. The information to be exchanged is packaged in periodically transmitted messages[7], as defined in [24]. Figure 2.1 shows the structure of a single CAM, which contains the following fields and containers:

- **ITS PDU Header**: this includes the version of the protocol used, the type of message being sent and the current identification number[8] of the transmitting station.
- **Generation DeltaTime**: this refers to the amount of milliseconds elapsed between the UTC-Start of 2004 and the CAM generation.
- **Basic Container**: this container includes the type of the transmitting entity and the geographical position at the time of the CAM generation.
- **High-Frequency Container**: this contains fast-changing data elements such as speed, heading and acceleration.
- **Low-Frequency Container**: this optional container carries slow-changing information, such as the role of the vehicle, the exterior lights and the latest path points.
- **Special Vehicle Container**: this container is reserved for vehicles with special roles, e.g. emergency vehicles, and contains information specific to the respective role.

[6] In this context, the equivalent type of message defined by SAE is the Intersection Collision Announcement [23].
[7] Based on the criticality of the situation around the transmitting entity, CAMs are transmitted with a frequency between 1 and 10Hz, while BSMs have a fixed transmission frequency of 10 Hz.

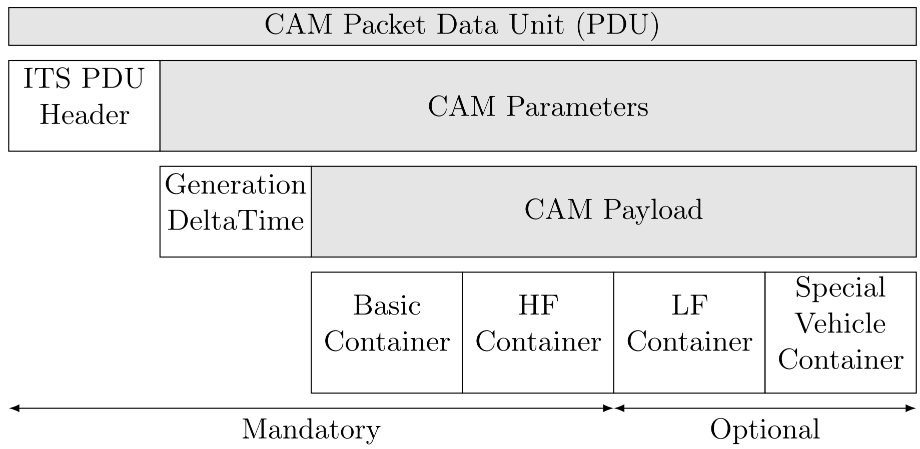


Figure 2.1: Structure of a Cooperative Awareness Message (CAM).

PDUs of CAMs are digitally signed and accompanied by the corresponding certificate, in order to ensure the integrity of the transmitted data. The usage of those certificates is further explained in Section 2.1.3.

The Basic Container and the High-Frequency Container are the most relevant containers for the detection and prevention of collisions, as these are both mandatory containers, which carry fast-changing information about the state and dynamics of the vehicle. On this work, the elements on the Low-Frequency Container are not used for the prediction of collisions, as this container is not mandatory and there is therefore no guarantee that it is going to be implemented and sent by all V2X-capable vehicles. For a detailed list of the data elements contained in the different CAM containers, please refer to Appendix 7.1.

It is also worth noting that the size of CAMs can vary between around 200 bytes and 800 bytes depending on their contents [26]. This increase in size[9] is mainly due to the certificates and signatures, which are typically between 100 and 150 bytes long, and the optional Low-Frequency Container, whose data element *pathHistory* can be up to 560 bytes long. Nevertheless, certificates and/or Low-Frequency Containers are not included in every transmitted message, i.e. they are sent only once per second, thus being the average size for CAMs 350 bytes [26].

[8] The ID of the entity is a pseudonym, i.e. it changes over space and time [25].
[9] The size of all mandatory data fields of CAMs and BSMs does not exceed 70 bytes.

2.1.3 Public Key Infrastructure (PKI)

As on almost every area of our society, the communication between vehicles will not be exempt of threats by entities wanting to tamper the flow of information. The communication scheme should hence be secured in order to guarantee two important points:

- Integrity of the data, i.e. protection against data manipulation [27].
- Confidentiality and end-user privacy, i.e. protection against misuse of information [16].

The deployment of a PKI for V2X-communications is part of the solution for guaranteeing these points. The architecture of this infrastructure is depicted in Figure 2.2. Under this scheme, vehicles dispose of digital certificates for signing outgoing messages and for verifying incoming messages. This aim to ensure that every communicating entity is appropriately authenticated and that the messages sent cannot be modified by third parties.

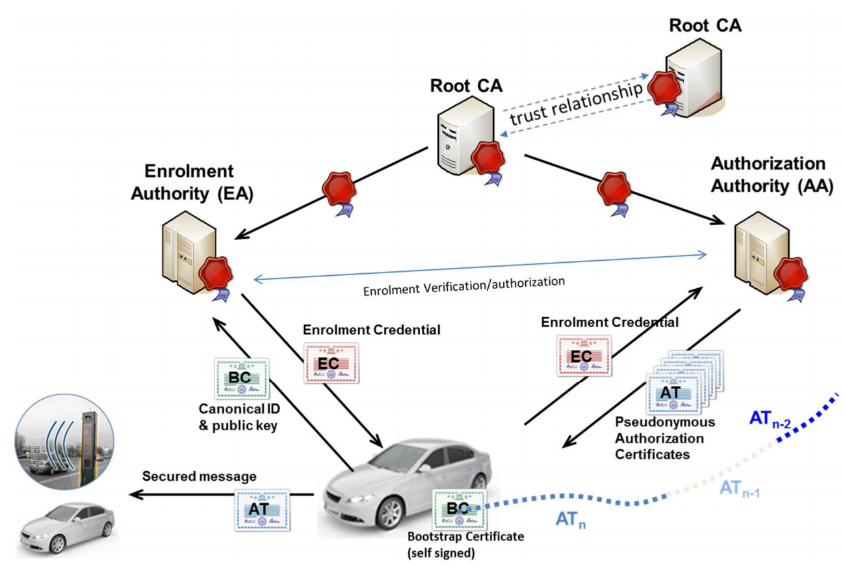


Figure 2.2: ITS Trust Model. Architecture of the PKI proposed by ETSI [28].

One of the challenges that arises from the implementation of this infrastructure is the verification of signatures, which are created using the ECDSA standard [28]. This verification can be very time consuming and could lead to a delayed intervention of the safety applications. For this reason, many companies and institutions are making efforts to reduce the time needed for signature verifications[10]. The hardware used for the implementation of this work in real vehicles includes a commercial-available communication processor, which is able to verify more than 2300 ECDSA signatures per second for the *NIST P256* curve [29].

[10] The time needed for the signing process is not as critical as the one for the verification, as data packets are sent at a maximum rate of 10Hz, i.e. every 100ms, while several of messages can be received within one second.

Pseudonyms

Digital signatures help authenticate the source of V2X-messages, but in order to preserve the anonymity and privacy of end-users, messages are signed using temporary identifiers called Authorization Tickets (ATs). Every vehicle can have up to 100 ATs simultaneously and must switch them based on the time elapsed and the distance travelled since the last change. Every AT has a validity period of 1 week, and can be obtained 3 months before the validity period starts [30].

A vehicle could then change its identifier at any moment[11], which improves the privacy of end-users but at the same time hinders the association of received messages and nearby-vehicles[12]. Out of this security feature arises the need of computing the probability of collision by using a single received CAM.

2.2 Localization

Vehicular localization is a crucial component for V2X-based solutions to work reliably and efficiently. The location of vehicles is mainly obtained through satellite-based systems, e.g. GPS, GLONASS, GALILEO and Beidou, as the infrastructure for their utilization is already established and worldwide available.

Because of the direct utilization of the signals received from the satellites only providing a limited accuracy, i.e. Standard Positioning Service (SPS), methods were developed in order to increase the accuracy of satellite positioning systems. Those methods include augmentation techniques based on the use of reference stations, e.g. Differential Global Positioning System (DGPS), as well as others based on the use of inertial sensors, e.g. Dead Reckoning. Table 2.2 lists some of the station-based augmentation techniques, comparing their accuracy with that of the SPS.

The level of accuracy needed by a V2X-application to operate reliably mainly depends on the nature of the application[13]. In the context of vehicular localization, three main different levels of accuracy can be distinguished [32]:

- **Road accuracy**: this level of accuracy can clearly identify the current road, this requiring an accuracy of at least 5 meters.
- **Lane accuracy**: the system is able to know on which lane of the road the vehicle currently is. An accuracy of 1.5 meters or better is needed for achieving this level of accuracy.

[12] This change could also be followed by a random silent period of up to 13 seconds [17].

[12] Some works about the impact of pseudonyms on safety-related applications show that the pseudonym change-policy negatively affects the ability of applications for predicting collisions [16]. Their recommendation is to wait for safe situations before changing pseudonyms.

Method	Accuracy	Range
SPS	8-15m	-
DGPS	0.5-5m	$<$ 100km
LADGPS	1-3m	$<$ 100km
WADPGS	0.5-2m	$>$ 100km
RTK	0.02-0.2m	$<$20km
RTK	$<$0.02m	$<$10km

Table 2.2: Accuracy of augmentation methods for GNSS receivers [8, 31]. Column *Range* refers to the distance from the nearest reference station.

- **Sub-lane accuracy**: having a sub-meter accuracy, systems in this category can distinguish in which part of the lane the vehicle currently is.

Although centimeter-level accuracy having the potential of greatly increasing the reliability of vehicular applications, the cost of such GNSS-receivers is comparable to the cost of middle-class cars. This limits their usage to reference systems and is the reason why they are not installed in mass-produced vehicles.

Localization requirements

The accuracy level considered for this work is *Road accuracy*[14], as this can be reached by most commercial-available, automotive-grade modules. The use of this accuracy level is also considered to be realistic, as the main application field for V2X-applications are urban environments, and such environments considerably restrict the precision of positioning systems.

And it is also worth noting that safety-related V2X-applications can only benefit from a highly-accurate position when all vehicles involved have a comparable accuracy. This is another reason for selecting the *Road accuracy* level as the baseline for the proposed solution.

[13] Applications such as *Blind-Spot Warning* require a much higher precision regarding vehicle localization than applications like *Approaching Emergency-Vehicle Warning*.
[14] The localization module used for the implementation of this work has a Circular Error Probability (CEP) of 2.5 meters and uses Dead Reckoning for improving the measurements in areas with poor signal reception [33].

2.3 Vehicle Data

As already mentioned, the information regarding the surrounding entities is gathered through the received CAMs. In contrast, the information corresponding to the Ego-vehicle is found locally in form of data packets that are sent by different controller devices through a bus network. This local information is also used for generating the outgoing CAMs.

On V2X-capable vehicles, a possibly distributed Controller Device is in charge of the vehicular communication. It is responsible for managing oncoming and outgoing V2X-messages, collecting local and remote data, and executing the V2X-applications. The following are the sources of information for such a Controller Device:

- **GNSS Receiver**: This module can be directly contained in the Controller Device or connected to it via the bus network. Its main tasks are the calculation, filtering and conditioning of the vehicle's geographical position.

- **Fixed information**: This refers to non-volatile information which is stored in the Controller Device and includes information related to the vehicle which does not change over time, e.g. vehicle dimensions, station type and role of the vehicle[15].

- **CAN-Bus**: This is a vehicle bus standard that allows devices to communicate with each other without the need of a master or host computer. Every device connected to the bus sends broadcast messages, which are received by all other devices on the bus. The following are two crucial devices, whose outputs could be present on the CAN-Bus:

  - **Speed sensor**: typically a part of the Anti-lock Braking System (ABS), this sensor measures the angular velocity of the wheels and computes a speed value for the whole vehicle. Due to their robustness while operating in harsh environments, the speed of every wheel is usually measured using inductive sensors.

  - **Inertial Measurement Unit (IMU)**: this sensor uses accelerometers for measuring the forces acting on the body and gyroscopes for measuring the angular velocity on every axis. Figure 2.3 shows the axes measured by a 6-DOF IMUs.

  At least one CAN-Bus is present in all modern vehicles and principally targets the exchange of information of powertrain and drive electronics. But due to its robustness, capability of simplifying electrical connections, always more sub-systems are connected to this bus.

[15] The possible station types correspond to pedestrian, cyclist, motorcycle, passenger car, bus and truck, among others, while the possible vehicle roles include taxi, public transport, emergency vehicle, military vehicle, agricultural machinery, etc. [25].

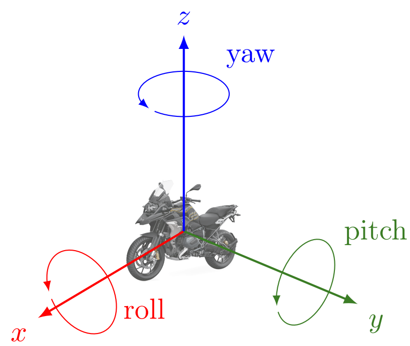


Figure 2.3: Axes measured by a 6-DOF Inertial Measurement Unit. The *x-axis*, *y-axis* and *z-axis* correspond to the longitudinal, lateral and vertical acceleration of the vehicle, respectively. The *roll* component plays an important role on motorcycles, as it allows the computation of the lean angle.

- **LIN-Bus**: This standard describes a cost-effective solution for building sub-networks targeting automotive comfort functions and body electronics. It can be seen as an inexpensive version of the CAN-Bus, which is used for the communication with non-safety-relevant devices and other sub-systems. A relevant information that can be found on this bus is the state of the exterior lights, which includes the state of the turn signals.

Table 2.3 presents a summary of the common sources for the data elements required by the proposed V2X-application and also used for generation of CAMs.

	Controller Device	GNSS Receiver	CAN-Bus	LIN-Bus
Time	✓	✓		
Position		✓		
Heading		✓	✓	
Speed			✓	
Driving direction			✓	
Acceleration			✓	
Curvature	✓			
Yaw-rate	✓		✓	
Exterior lights			✓	✓

Table 2.3: Sources of vehicle-related information.

2.4 Digital Map

One of the methods for increasing the contextual awareness of vehicles is the inclusion and usage of digital maps. A digital map refers to the discretized ensemble of data representing a road structure, and containing detailed information about the elements on it.

Modern vehicles use digital map data mostly for improving the driving experience in the form of visual representations of the road ahead, recommendations about the route to follow and information about road characteristics, e.g. speed limits. But a digital map can also be viewed as an additional sensor, having the potential of improving the performance of safety applications. Their information can be used as it is, i.e. the digital map acts as a primary sensor, or it can be used for validating or enhancing the data coming from other sensors [34]. The function of digital maps in this work belongs to the latter, i.e. it acts as a secondary sensor acting as a database for enhancing the capabilities of the V2X-application.

Three main types of digital maps for vehicles can be distinguished according to their accuracy level and the features they offer[16], this combination would be hereafter referred to as the *quality level* of a map. Table 2.4 gives an overview of the features associated to different quality levels of digital maps.

- **Standard maps**: those topological maps mainly target navigation and route planning. They include a relative basic, scaled road topology, which must not (or cannot) be frequently updated for its operation. Examples for this quality level are the built-in maps on older vehicle models. They can work as standalone, primary sensors, as they don't need to communicate with the rest of the system. Printed maps and satellite images are commonly used for the creation of such maps.
- **Enhanced maps**: those are an enhanced version of standard maps that, besides being topological and able to handle navigation tasks, serve as a source of information for other sub-systems. They have an improved accuracy, generated in a finer scale, and a larger set of features and elements. An example for this category is the map created by the OpenStreetMap (OSM) project.
- **High-Definition maps**: mainly targeting applications in the scope of autonomous driving, those topographical maps have a centimeter-level accuracy and a much higher amount of features than standard and enhanced maps. Examples for this quality level are the maps offered by the company HERE. Those maps are commonly generated by terrestrial mapping systems instead of by humans [34].

[16] Different map providers include a different set of features in their maps. The features here mentioned correspond to the ones usually supported by the digital maps on the different categories.

Feature/Element	Quality level		
	Standard Map	Enhanced Map	High-Definition Map
Road topology	●	●	●
Road category	●	●	●
Road signs	-	◐	●
Road curvature	-	◐	●
Road steepness	-	-	●
Coordinates of intersections	●	●	●
Signal-controlled intersections	-	●	●
Right of way at intersections	-	◐	●
Amount of lanes	◐	●	●
Lane width	-	◐	●
Allowed maneuvers on lanes	-	◐	●
Markings over the lanes	-	-	●
Lane delimiters	-	-	●
Fixed speed limits	●	●	●
Variable speed limits	-	◐	●
Live updates (internet access required)	-	◐	●
Elements as reference for positioning (3D landmarks)	-	◐	●

● = supported; ◐ = partially supported; - = not supported

Table 2.4: Digital map features according to their quality level. Features and elements are divided into four categories here: road characteristics, intersection characteristics, lane characteristics and other characteristics.

Digital map requirements

The following are the intersection-related information to be used as an assistive resource by some of the presented approaches:

a) Geographical coordinates of intersection: latitude and longitude corresponding to the center of the intersection.
b) Size of intersection: equivalent to the number of lanes of the intersecting roads.
c) Type of road: used for estimating the lane width, if not directly given, and including categories such as highway, avenue, residential, rural road, etc.
d) Right-of-way: presence or absence of traffic signs regulating the flow of traffic.
e) Presence of traffic light: used for the identification of signal-controlled crossings, which could work in cooperation with other V2X-applications able to process information sent by traffic lights.

Area	Size	Access-time
Europe	21.7 GB	100 %
Germany	3.2 GB	80 %
Bavaria	581 MB	50 %
Upper Bavaria	170 MB	38 %

Table 2.5: Size and access time for different map areas. Those sizes correspond to .osm.pbf files available on `https://www.geofabrik.de/`. The access time intends to show the relation between the size of an enhanced map and the time needed for extracting data out of it. Those times are computed by running an overpass-command for finding the ID of the current road based on the current geographical position.

The required quality level for the here presented application is that of *Enhanced maps*, as this level contains all the required information and its size is much more manageable than the one of High-Definition maps[17]. Table 2.5 lists the size and relative access-time corresponding to an *Enhanced map* covering different areas.

Additionally, as the selected digital maps are going to be used by safety-critical applications, it is important that they respond to queries with a very low latency [14]. Because of this, they should ideally be available offline, in order to not having to wait for a response coming from network-dependent remote servers.

2.5 Human-Machine Interface

Because of human beings and vehicles delivering and responding to different kinds of stimuli, an interface is needed so that they can understand each other. This being the task of Human-Machine Interfaces (HMIs).

Modern HMIs intend to provide only the necessary information at the right time, in order not to distract end-users from attending their main task, i.e. safely operating the vehicle. Out of this need of only providing situation-relevant information, HMIs manage the information to be given by prioritizing, filtering and scheduling [36], and don't always communicate with the end-user via the same communication channel (visual, auditive, haptic, etc.), nor with the same emphasis (static/blinking, silent/loud, soft/pronounced, etc.), nor at the same place (mirrors, cockpit, windshield, helmet, etc.). The form of the communication is selected depending on how relevant the information actually is for the end-user.

[17] High-Definition maps require a very large memory space, usually in the order of Terabytes [35].

HMI requirements

V2X-applications rely on HMIs for acting as fast as possible and correctly reflecting the criticality of the situation, so that the end-user clearly recognizes which actions need to be taken. HMIs should then be able to issue information to the end-user using at least two levels:

- **Notifications**: those signals have an informative nature, target relevant (but not necessarily critical) situations and are strategically place for only attracting the attention of the end-user when the transmitted information is needed. A good example of this information level is a modern Blind-Spot warning, shown in Figure 2.4.

Figure 2.4: Example of a notification given by a HMI. The shown HMI corresponds to a *Blind-Spot Monitor* on a BMW Hybrid Active 3. This is a sensor-based solution which detects the presence of vehicles located in its blind-spot. When an intention of changing lanes is detected by the system, a yellow triangle is illuminated for making the end-user aware of the other vehicle. Image source: [37].

- **Warnings**: those are acute signals, which intend to attract the attention of the end-user and indicate that an urgent action is needed. They are only to be issued when a critical situation is detected, and also early enough so that a collision can still be avoided. Figure 2.5 shows an example of such a warning.

Those two levels of information correspond to the ones proposed by CMC and implemented in this work, namely Motorcycle Approach Indication (MAI) and Motorcycle Approach Warning (MAW)[18]. The former intends to increase the awareness of road users about the presence of nearby motorcycles, while the latter aims to influence the behavior of end-user by inducing an immediate reaction. Table 2.6 gives an overview of those levels of information. It should be noted that a single application can implement both levels simultaneously, i.e. it is able to provide both notifications and warnings.

[18] In the scope of CMC, those are not only information levels, but also categories grouping several V2X-applications.

Figure 2.5: Example of a warning given by a HMI. The shown HMI corresponds to a *Forward Collision Warning* on a VW Passat. This sensor-based solution measures the distance to the vehicle ahead and uses the difference of speeds for estimating if a collision is probable. If it is, a visual warning, usually accompanied by auditive and/or haptic signals, is given. Image source: [38].

Level	Provides	Based mainly on	Situation	Action expected
MAI	Information	Position of vehicles	Relevant	Increased attention
MAW	Warning	Position and movement of vehicles	Critical	Braking / Steering

Table 2.6: Comparison between the communication levels MAI and MAW.

Additionally, the information issued at the MAI and MAW levels should be clearly distinguishable (visually, acoustically and haptically) and, as several V2X-applications could operate concurrently, their informational output should be prioritized in order not to overload the end-user[19]. Further aspects to be taken into consideration during the design of appropriate HMIs for V2X-applications are presented in the Annex D of [22].

[19] Information should be prioritized and filtered according to both the intrinsic criticality (warnings have preference over notifications) and the time left for their occurrence.

3 Conception and Specification of IMA

3.1 Description

The Intersection Movement Assist (IMA) is a V2X-application able to warn the driver of a vehicle or motorcycle when it is not safe to enter an intersection due to a high probability of collision with other vehicles.

This application uses the vehicle data and the CAMs being sent by vehicles in its vicinity for computing the probability of a collision at an intersection. If necessary, this information is then given to the HMI, which in turn issues a notification or warning to the end-user in order to reduce the risk of a collision. The working environment of this application are intersections at which the paths of the Ego-vehicle and the Other-vehicle cross each other; this scenario receives the name of cross-traffic[1].

In order for IMA to work, all vehicles involved should be able to send CAMs, while the Ego-vehicle (having the IMA-application) must also be able to receive CAMs in order to process them. This highlights again the need for a wide dissemination of V2X-communication on vehicles, for which end-user's acceptance plays a crucial role.

Because of being a safety-critical application, IMA continues to work at signal-controlled intersections and junctions[2], regardless of which lights are green. It can also make use of different methods for getting a better picture of the situation and behaving accordingly, this includes information about the type and topology of the road, the applicable road regulations, e.g. right-of-way, presence of traffic lights or road signs, and more.

[1] In cross-traffic scenarios, the vehicles are not driving on the same road, but their trajectories cross at an intersection or junction.
[2] This refers to the presence of a working traffic light at the intersection or junction.

3.2 Use Cases

There are several situations at intersections that have the potential of becoming critical, specially when a driver without right-of-way[3] cannot stop on time, e.g. due to over-speeding, or oversees another vehicle having right-of-way, e.g. due to distractions, obstacles on the road or weather conditions. The following use cases illustrate some of those common situations, which shall be covered by this V2X-application.

- **Straight crossing paths**: two vehicles are approaching the same intersection and plan to drive across it.

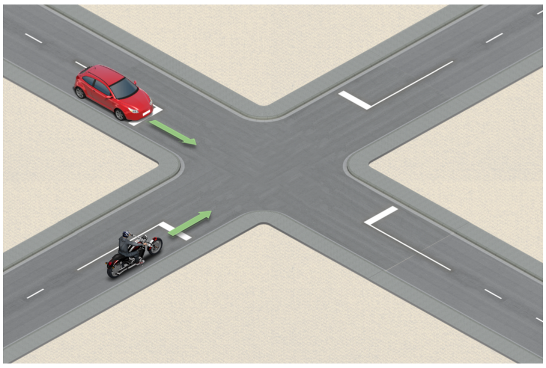


Figure 3.1: Use case *Straight crossing paths*[4].

- **Left turn into crossing**: a vehicle approaching an intersection and intending to turn left intersects the driving paths of other vehicles coming from the sides.

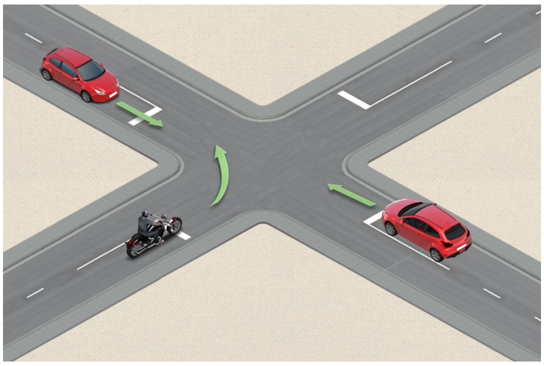


Figure 3.2: Use case *Left turn into crossing*.

[3] Right-of-way given by a traffic light, road sign or by the absence of those (right-before-left rule).
[4] This and the following use-case pictures are made using the Illustration Toolkit provided by the CAR 2 CAR Communication Consortium (C2C-CC).

- **Other-vehicle turning left from right**: here, a vehicle intends to travel across an intersection, while the Other-vehicle plans to turn left at the same intersection, and this way intersecting the driving path of the first one.

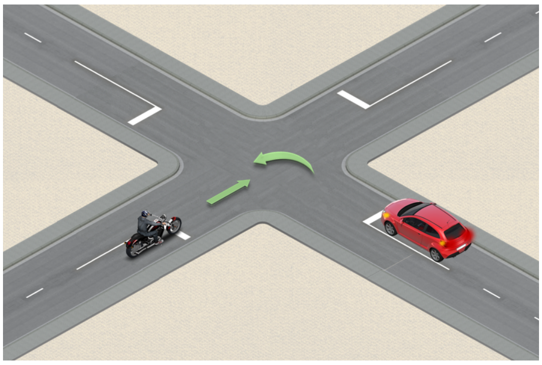


Figure 3.3: Use case *Other-vehicle turning left from right.*

- **Other-vehicle turning right from right**: in this scenario, a vehicle intends to cross an intersection while another vehicle turns to the right intends to turn right at the same intersection, resulting in an overlap of the trajectory of both vehicles.

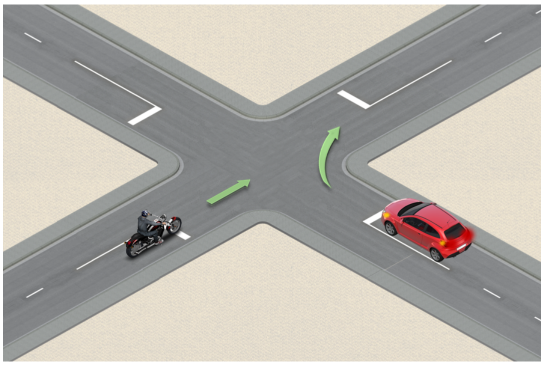


Figure 3.4: Use case *Other-vehicle turning right from right.*

3.3 Concept of Notifications and Warnings

As mentioned before, IMA requires the implementation of at least two information levels: notification and warning. Figure 3.5 shows a timeline illustrating the issue of information through the HMI before a collision occurs.

In this context, notifications are conceived to be given earlier as warnings and should lead to an increment of attention from the end-user. A warning, in contrast, should bring the end-user to perform an evasive maneuver in order to avoid the collision, i.e. brake and/or steer.

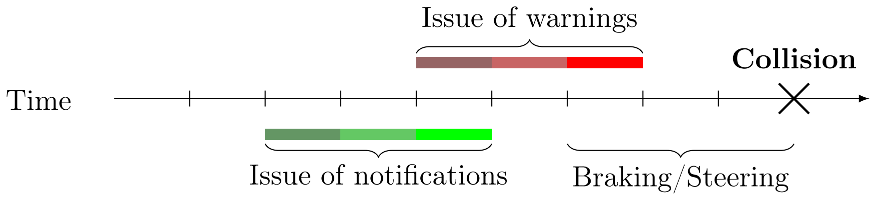


Figure 3.5: Timeline for notifications and warnings. The coloured rectangles correspond to different possible timings for issuing a notification or warning. Both notifications and warnings should be issued early enough, so that the end-user has enough time for avoiding the accident; this includes the perception of the notification or warning and the appropriate reaction, i.e. braking and/or steering.

At best, a notification would be enough for neutralising the situation, so that the collision is avoided without a warning needed to be given. But if that is not the case, the driver would also receive a warning, and in order to avoid overwhelming the driver with notifications promptly followed by a warning, a minimum separation of 1.5 s between those should be maintained [39].

Aiming at increasing the system acceptance, map data could be used for only giving notifications to the participants without right-of-way[5]. This measure aims to increase the attention of those, which must give way to other road users, while not distracting the end-users having right-of-way, as such a notification not being of great use for them.

3.4 Requirements

In order to work properly, be robust and achieve a suitable level of acceptance, IMA must be able to fulfill several requirements. This section deals with the different categories of requirements to be fulfilled by this application. A summary of those is presented in Table 3.1.

3.4.1 Activation Requirements

The application IMA is not conceived for being constantly active[6], but for computing the probability of a collision only after some prerequisites are met. This measure exclude scenarios and situations, which would not lead to the issue of a notification or warning, thus sparing computation resources and time.

[5] All participants would still be able to receive warnings, independent of right-of-way, because of warnings not being informative only, but rather a request to take action.

Needs	Requirements
IMA needs to know when a collision at an intersection involving the Ego-vehicle is about to take place.	IMA shall locally acquire location and motion parameters of the Ego-vehicle, in order to compute the risk of a collision.
	IMA shall acquire location and motion parameters of the Other-vehicle through CAMs, in order to compute the risk of a collision.
IMA needs to make sure that the information coming from other entities is valid and relevant, before using it for issuing notifications or warnings.	IMA shall verify that every received CAM contains a valid certificate or that it was sent by an entity with an already known certificate (see section 3.4.2).
	IMA shall control that the received CAMs fulfill a series of prerequisites before proceeding to the risk computation (see section 3.4.1).
IMA needs to detect when components of the system fail.	IMA shall notify end-user, via the HMI, in case the application is not working properly (see section 3.4.4).
IMA needs to detect and inform about critical situations, giving the end-user the possibility of avoiding them.	IMA shall process incoming messages and issue information to the end-user, if necessary, within a given time-frame (see section 3.4.3).
	IMA shall provide notifications and warnings to the end-user over a HMI.

Table 3.1: Summary of requirements to be met by the application IMA.

- **Speed range**: the speed of the Ego-vehicle should be equal or lower than 100 km/h, as this is the highest speed allowed in rural areas, according to German traffic regulations (§3 Par. 3 of StVO). The speed limit for rural roads in other EU Member States also lies under this threshold [40].

- **Location**: an intersection or junction is located ahead of the Ego-vehicle and the current road type does not correspond to a motorway with a speed limit above 100 km/h. This is only applicable when map data is available.

- **Driving direction**: the headings of the Ego-vehicle and Other-vehicle indicate that they are approaching to each other and that their trajectories cross each other at an intersection[7].

- **Distance to Other-vehicle**: Only the CAMs from entities closer than 300 m are to be considered[8] [22]. This enables the detection of vehicles on time even when driving at a relatively high speed.

[7] When not using map data, and in order to consider the prerequisite as fulfilled, the trajectories must be found to cross each other transversely.

[8] For an intersecting angle of 90 degrees, the distance from the vehicles to the intersecting point corresponds to 212.13 meters, which is also be the minimum distance to the conflict area for at least one of the vehicles.

3.4.2 Security Requirements

As mentioned in section 2.1.3, all transmitting entities use digital certificates for signing outgoing messages in order to demonstrate that they are entitled to transmit those messages. The signing certificates are not present on every outgoing CAM, but are transmitted at least once every second [24]. All V2X-messages not containing a certificate include a digest of the certificate previously sent in order to reduce the overall message length while still allowing all V2X-messages to be verifiable.

Before giving any notifications or warnings to the end-user, the application should then verify that the information used for the risk computation is valid comes from a trusted source. This includes the verification of the following:

(a) the message payload is consistent with the permissions on the attached certificate.
(b) the signature matches the accompanying data and certificate[9].
(c) the signing certificate chains up to an already-trusted certificate or to the root certificate.
(d) all certificates on this chain are valid, i.e. the have not been revoked.

It remains open, however, if the verification of digital signatures should take place before or after computing the probability of a collision. And because of the time and computational effort needed for verifying digital certificates not being negligible, the relevancy of the message could be examined first.

3.4.3 Real-time Requirements

One of the most important requirements for this application to fulfill are the real-time requirements, because of a warning issued too-late being useless or even distracting for the end-user. According to previous works, and in order for these kind of warnings to be effective, the Maximum Latency Time (MLT) of the system must satisfy the following [14]:

$$\text{MLT} \leq 200\,\text{ms} \tag{3.1}$$

The MLT describes the time for processing the received V2X-messages, i.e. the total time for computing the risk of a collision, and the issue of information to the end-user over the HMI[10]. This MLT can be described as the sum of the following components:

$$\text{MLT} = t_{VehicleData} + t_{Cert} + t_{Risk} + t_{HMI} \tag{3.2}$$

[9] For validating the signature, this is decrypted using the public key attached to the digital certificate, and the resulting decrypted block of data must be equal to a hash of the received data, which is obtained using the hash algorithm *SHA-256* [41].
[10] MLT can also refer to the time until an automated response is initiated by the vehicle.

- Time for gathering information related to the current state of the Ego-vehicle ($t_{VehicleData}$): this includes reading data from the CAN-Bus, the LIN-Bus and the GNSS-receiver. This information is to be collected when generated instead of being gathered upon request. Which leads to low latency times ($\ll 1\,\text{ms}$), as up-to-date information would be already stored in the local memory of the Controller Device.

- Time for validating the digital certificate of the received CAM (t_{Cert}): according to the performance of commercial-available hardware [29], the following validation times can be expected for the indicated elliptic curves:

$$t_{Cert} = \begin{cases} \sim 0\,\text{ms} & \text{if Certificate already known}^{11}, \\ 0.43\,\text{ms} & \text{if ECDSA\_NIST\_P256 is used}, \\ 2.56\,\text{ms} & \text{if ECDSA\_NIST\_P384 is used}, \\ 1.37\,\text{ms} & \text{if ECDSA\_Brainpool\_P256 is used}, \\ 3.33\,\text{ms} & \text{if ECDSA\_Brainpool\_P384 is used}. \end{cases} \tag{3.3}$$

- Time for estimating the collision risk (t_{Risk}): during this time, the application uses the previously gathered information for estimating the probability of a collision. If map data is available, this also includes obtaining the intersection attributes from the corresponding database.

- Time for presenting warning over HMI (t_{HMI}): this is the delay between the application having having informed the system about the need of a notification or warning and the information being presented to the end-user. This delay strongly depends on the implementation of the HMI, but is usually located in the lower millisecond-range.

The application must also be able to process at least 1000 V2X-messages per second [22], which realistically corresponds to the maximum number of messages which can be received by a road entity within one second. This maximum number of messages per second is based on the time needed for the transmission through the physical channel and is bound to the communication technology being used. In the case of the C-V2X-technology, a sub-frame for the transmission of a whole message lasts 1 ms (see appendix 7.2).

This means that in order for the application to consider all incoming information, it must be able to process CAMs with time

$$t_{Risk} < 1\,\text{ms} \tag{3.4}$$

[11] As the same certificate is used as long as the pseudonym does not change [17], it can be verified only the first time and then stored as such in local memory in order to avoid the need for verifying it again.

3.4.4 Reliability Requirements

The reliability of the application can be divided into two main aspects: the first one is the reliability of the information given, i.e. how accurate is the algorithm at detecting collisions, while the second one refers to the level of fault-tolerance of the system.

Regarding the reliability of the information given, the application should aim to minimize the below-mentioned factors:

- False positives (or false warnings), defined as the warnings given in non-critical situations.
- False negatives (or missed warnings), defined as the critical situations for which no warning is given.

Those factors can be used for describing the performance of the collision-detection algorithm and their minimization leads to an increase in acceptance from the end-user [13]. A list with all the performance indicators used in this work is found in section 5.3.

Regarding the aspect of fault-tolerance, the application must be able to detect when components fail or are offline. It should then enter a fail-silent mode and inform the end-user about this state. While in fail-silent mode, received V2X-messages are not processed and no notifications or warnings are given to the end-user.

The application could also perform a self-check while keeping an indicator light turned on until this self-check is finished, thus operating similar to the diagnostic self-check of Anti-lock Braking Systems on modern motorcycles[12].

Such a self diagnostic should at least confirm that the following elements are working properly:

(a) **Communication**: the system is able to receive CAMs from another road entity.
(b) **Security module**: received CAMs can be validated according to 3.4.2.
(c) **Bus system and sensors**: all necessary vehicle data is available.
(d) **Collision-detection algorithm**: the probability of a collision can be computed.
(e) **Information output**: communication with the HMI can be established.

3.4.5 Compatibility Requirements

The information contained in the received V2X-messages should comply with current standards [25, 42]. And although the specification of this V2X-application is mostly based ETSI's standards, all data elements used by the algorithms in this work are to be found in both BSMs and CAMs, which makes possible using the same collision-detection algorithms for both standards.

[12] This system waits for the motorcycle to start moving, in order to check that all sensors are working properly. The indicator light remains consequently on, until these checks are done.

3.5 Approaches

This section presents different approaches for the implementation of a collision-detection algorithm based on V2X-communication.

The following story-line describes a dangerous situation taking place at a road intersection, and will be used for illustrating how the different approaches work. As seen in Figure 3.6, four different vehicles are located near to an intersection, three of them (EV-1, OV-1 and OV-2) are driving into the intersection while the fourth one (OV-3) is getting away from this.

Both OV-1 and OV-2 are driving into the intersection and have right-of-way over EV-1. OV-2 is driving at a relative low speed while OV-1 drives at an adequate speed for these road. EV-1 plans to cross the intersection and becomes aware of OV-2 and OV-3, but not of OV-1 because of OV-3 interrupting the Line-of-Sight between EV-1 and OV-1. Because of OV-2 driving slowly, EV-1 sees an opportunity for crossing the intersection. But as it tries to drive across the intersection, EV-1 collides with OV-1 and alleges not having seen OV-1 coming.

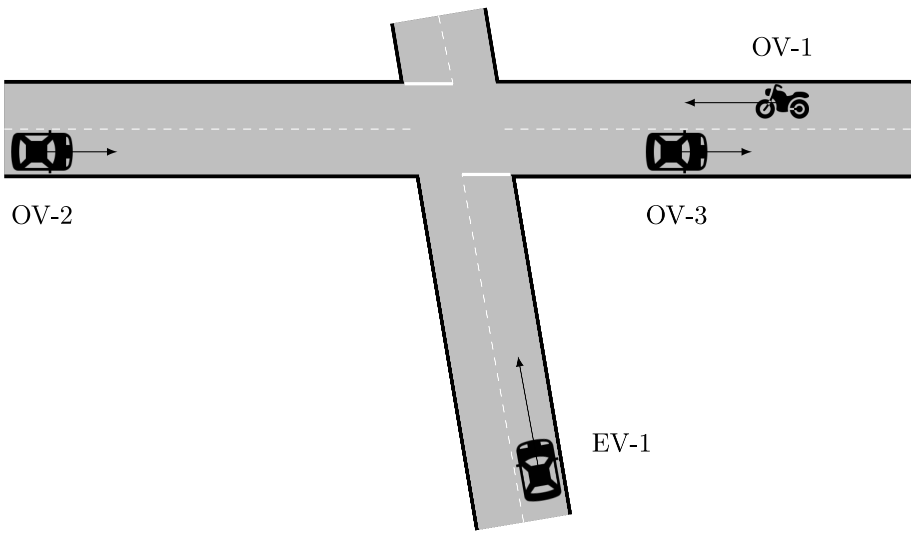


Figure 3.6: Example of a dangerous situation at an intersection. EV corresponds to Ego-Vehicle and OV corresponds to Other-Vehicle.

This kind of situations can be avoided by using V2X-communication together with a collision-detection algorithm. The following subsections present different approaches for the implementation of such an algorithm. It is assumed that all vehicles involved in this scenario are capable of broadcasting V2X-messages, and that only the Ego-Vehicle (EV-1) makes use of a collision-detection algorithm.

3.5.1 SAE-based

This approach is based on the document *J2945/1* from the Society of Automotive Engineers (SAE), which describes on-board system requirements for V2V safety communications [42]. It makes use of two relevant zones for the detection of vehicles which could pose a threat for the Ego-vehicle. Figure 3.7 shows the above described situation under the implementation of this approach.

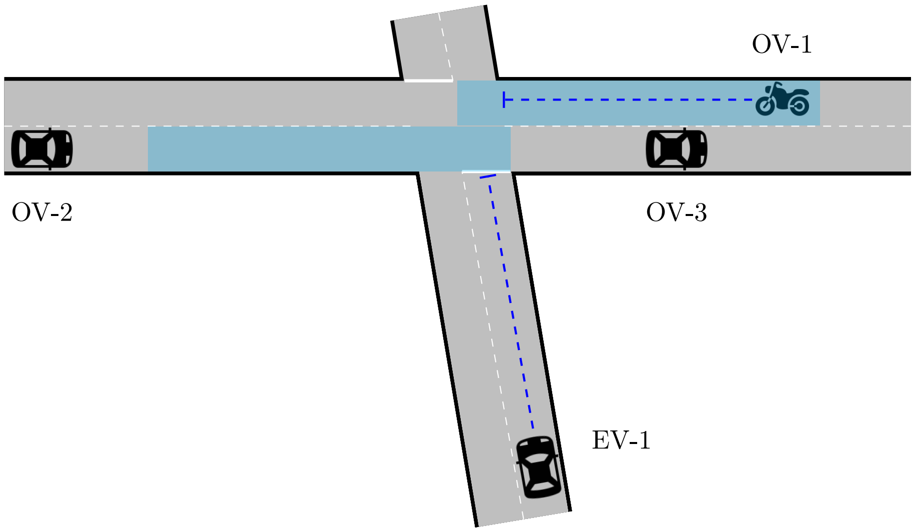


Figure 3.7: Concept of collision detection at intersections proposed by SAE. The blue rectangles represent the relevant zones to be monitored by the application. The blue, dashed lines represent the Time-to-Intersection (TTI) and Distance-to-Intersection (DTI) for EV-1 and OV-1.

Two relevant zones are projected into the road and the vehicles inside of them are taken into consideration for computing the collision risk. Vehicles inside of these zones are then near enough to the intersection ahead of the Ego-vehicle, and also driving towards this intersection, because of the covered lanes corresponding to lanes going into the intersection. As appreciated in Figure 3.7, only one of the three OVs is inside a relevant zone. This is because of OV-3 having passed the intersection and OV-2 still being too far away from it.

In order to avoid the collision between EV-1 and OV-1, this approach would compute the probability of them being inside the intersection at the same time. For doing this, a Time-to-Intersection (TTI) and a Distance-to-Intersection (DTI) is computed for every vehicle inside of a relevant zone and then contrasted with the TTI and DTI of the Ego-vehicle in order to estimate the risk of a collision and give a warning, if needed.

3.5.2 ETSI-based

This approach is based on the document *ETSI TS 101 539-2* from the European Telecommunications Standards Institute (ETSI) [22]. This approach utilizes the concepts of Time-to-Collision (TTC) and Dynamic Safety Shield (DSS) for detecting potential collisions. This concept is depicted in Figure 3.8.

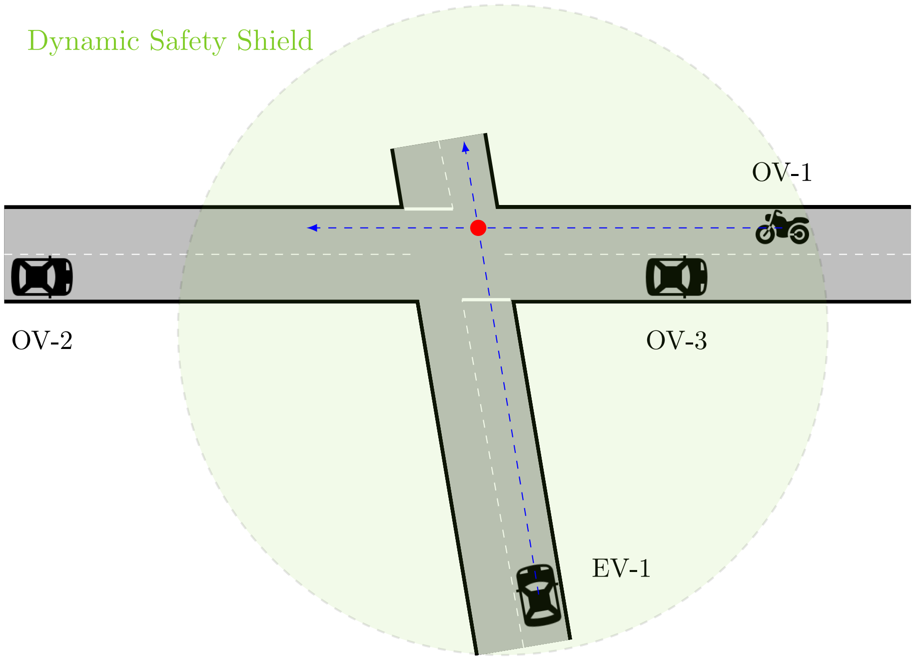


Figure 3.8: Concept of collision detection at intersections proposed by ETSI. The blue, dashed lines represent the trajectories to be followed by the vehicles and the red point is the intersection of those trajectories, representing the Point-of-Collision.

The DSS is a virtual area surrounding the Ego-vehicle. The form of this shield is flexible and all vehicles inside of it are considered relevant for the collision-detection algorithm. The TTC is the time for a vehicle to reach the estimated Point-of-Collision under constant driving conditions. In order to trigger a warning, this parameter should be below a threshold TTC_{min}, which indicates the time left for the driver to take an action and is specified as follows:

$$TTC_{min} > MLT + MDRT + MAT + \varepsilon \tag{3.5}$$

In this equation, the MLT denotes the maximum delay allowed when issuing a warning (see section 3.4.3), the Maximum Driver Reaction Time (MDRT) denotes the time elapsed between the warning being given and the driver taking an action, and the Maximum Action Time (MAT) indicates the time needed for a collision-avoidance action, e.g. the time for the vehicle to stop. For taking the error of the sensors into consideration, a time margin ε is added to the equation.

Under this approach, and in order to avoid the collision between EV-1 and OV-1, a Point-of-Collision (PoC) is calculated by using the driving trajectories of the vehicles inside the DSS. If there is no Point-of-Collision, as in the case of OV-3, the vehicle is not further considered by the algorithm. The obtained Point-of-Collisions are then used for estimating the TTC of every vehicle. A warning is given when both TTCs are similar, i.e. both vehicles are to reach the Point-of-Collision at approximately the same time, and the following holds for the Ego-vehicle:

$$TTC \leq TTC_{min} \tag{3.6}$$

3.5.3 ETSI-based with Map Data

This approach is based on the concept mentioned above, with the difference of having access to map data. By using this data, the algorithm can filter out areas which are not of interest for the application, e.g. parking lots, roundabouts, off-road, etc.

When using this approach, the trajectories of EV-1 and OV-1 are not needed for calculating the Point-of-Collision. This point is instead obtained by getting information about the position of the intersection and the lanes, on which the vehicles are currently driving. This Point-of-Collision is then used for the calculation of the TTCs needed for estimating the risk of a collision.

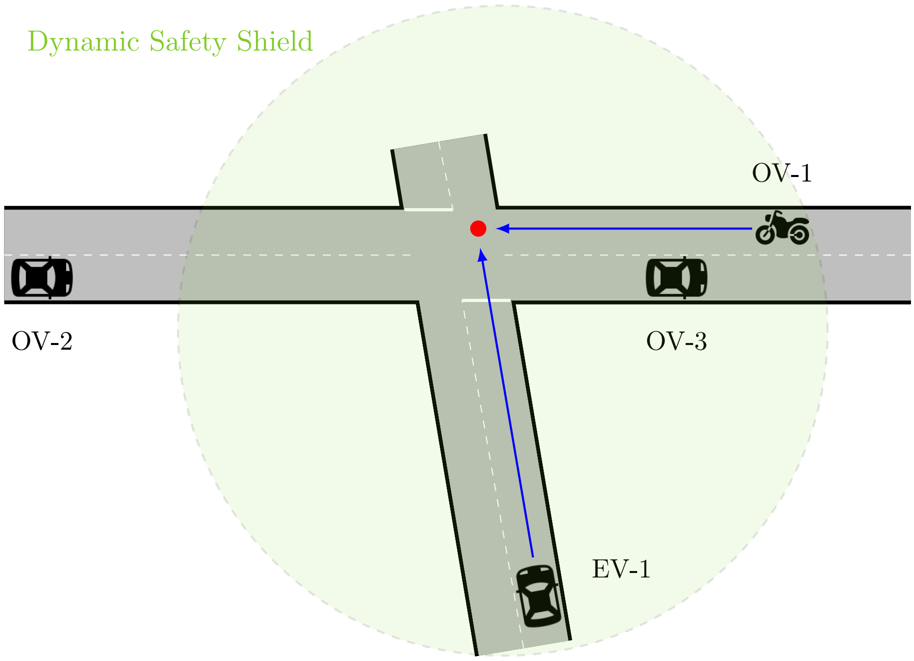


Figure 3.9: Map-assisted collision detection based on ETSI's concept. The red point represents the Point-of-Collision, while the blue arrows correspond to the time and distance to be covered by the vehicles in order to get to the Point-of-Collision.

3.5.4 Ghost Vehicles

This is an *a-posteriori* approach based on the idea of predicting the future position of the vehicles and then assessing if they overlap, which in turn indicates that a collision is to take place. An illustration of this concept is found in Figure 3.10. It shows multiple ghost vehicles representing the predicted position of every real vehicle.

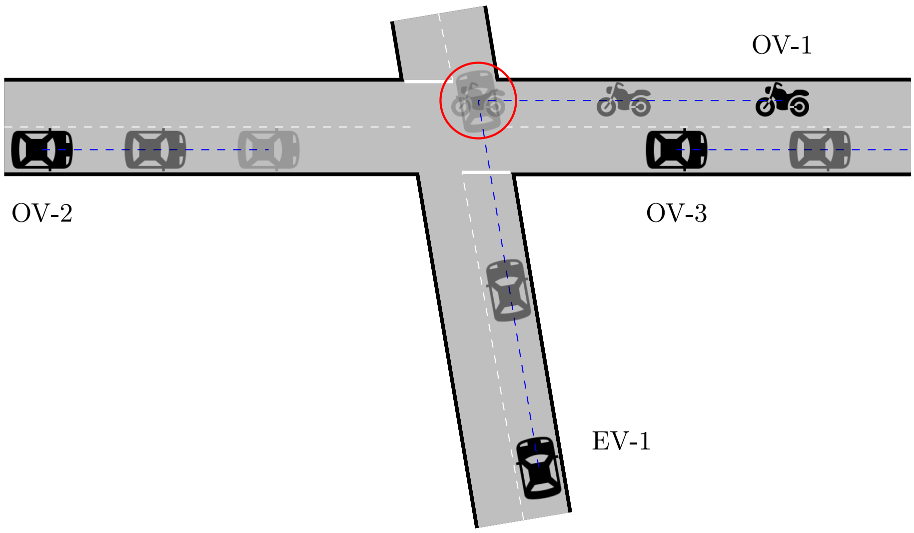


Figure 3.10: Concept of collision detection based on ghost vehicles. The semitransparent vehicles represent the ghost vehicles. Their level of transparency indicates how far into the future they are predicted. The red circle highlights the intersection between two ghost vehicles, which indicates that a collision is about to happen.

The future position of every vehicles is estimated according to its current driving state, i.e. position, speed, acceleration, etc., assuming a constant driving conditions and extending this driving state over a fixed amount of time. And every time, after the ghost vehicles are generated, the algorithm searches for overlaps between pairs of ghost vehicles by measuring the intersected area or the distance between the ghost vehicles.

For avoiding the collision between EV-1 and OV-1, this approach generates ghost vehicles for all vehicles in the vicinity of EV-1[13] and proceeds to measure the overlap between the generated ghost vehicles. If this overlap is found to be above a certain threshold, a warning is given.

[13] This approach does not make use of relevant zones or Dynamic Safety Shields, although those techniques could also be implemented for avoiding generating unnecessary ghost vehicles, e.g. for OV-3.

3.6 LTA as an Use Case of IMA

The Left Turn Assist (LTA) is another V2X-application which, due to its operation at intersections, could be seen as a sub-application of IMA. It covers oncoming-traffic scenarios and focuses on providing notifications and warnings to vehicle drivers intending to turn left at an intersection or junction. Thus preventing collisions caused by overlooking oncoming vehicles or incorrectly estimating their speed.

This application must recognize an intention of turning left, calculate the probability of a collision and transmit information to the end-user over the HMI, if the situation is found to be critical. The main difference between LTA and IMA lies on the kind of scenarios which are covered by them:

- LTA covers oncoming-traffic scenarios, i.e. both vehicles drive on the same road and in opposite directions.
- IMA covers cross-traffic scenarios, i.e. both vehicles drive on different roads, but their trajectories intersect each other[14].

LTA-specific Use Cases

- **Left turn with oncoming-traffic**: in this scenario, the Ego-vehicle (here a motorcycle) attempts to turn left at an intersection. The Other-vehicle approaches the same intersection from the opposite direction and plans to drive across it.

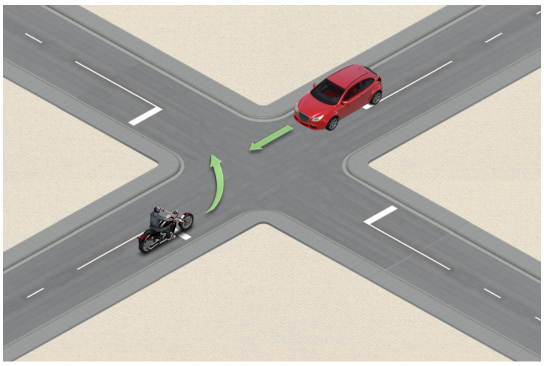


Figure 3.11: Use case *Left turn with oncoming-traffic.*

[14] These classifications are acknowledged by institutions such as ETSI and SAE [22, 42] and also found on existing sensor-based solutions targeting those scenarios.

- **Other-vehicle turning left ahead**: this scenario equates to the previous one, but with exchanged roles. Here, the Other-vehicle is the one intending to turn left at the intersection, while the Ego-vehicle attempts to travel across it.

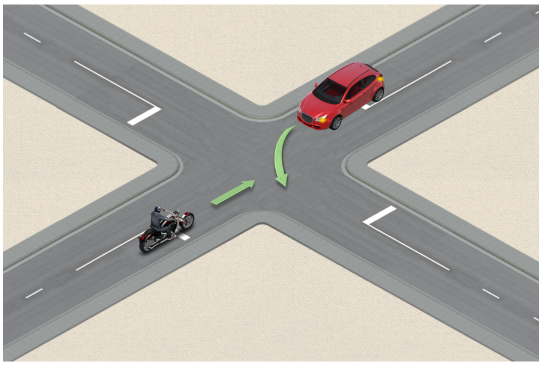


Figure 3.12: Use case *Other-vehicle turning left ahead.*

In order to stress the importance and the challenge of always being able to detect motorcycles, Figure 3.13 shows how external factors can make a motorcycle almost unrecognizable in an oncoming-traffic scenario.

Figure 3.13: Example of a motorcycle almost vanishing into the background. Even though the motorcycle has the headlight on, environmental conditions make the perception of it difficult for other road users. Image source: [43].

3.7 Discussion

This chapter presents a detailed description of the V2X-application IMA. The different use cases needed to be addressed by this application are illustrated and the different requirements to be fulfilled are also described.

Those requirements include the real-time characteristics in order for the application to be useful to the end-user. They specify the maximum time for the application to issue a warning, as well as the maximum time available for processing every incoming CAM. Complying with this processing time poses a challenge because of the time needed for the verification of digital signatures not being negligible.

Four different approaches for the implementation of this application are also presented. Three of these approaches are based on standards from different industry bodies while a fourth one uses a discrete paradigm for the risk estimation instead of a continuous one. An important concept used by all presented approaches is the TTC, which stands for the remaining time before a collision occurs. It has a lowest threshold, TTC_{min}, which is used to determine when a warning must be given to the end-user.

This chapter also illustrates an important separation between cross-traffic and oncoming-traffic scenarios. The former represents the scenarios being addressed by the application IMA while the latter describes scenarios to be covered by LTA.

4 Implementation

This chapter deals with the technical implementation of the different approaches for computing the probability of a collision. This implementation takes place in both a simulation environment and in prototype vehicles.

The first section covers the structure of the system and its different components, giving an overview of the data and modules needed for the implementation, while the next one explains the main tools and frameworks used for the realization of those modules. The following two sections explain in detail the parameters needed by the different algorithms and how they are handled by those. The last section then gives a summary of the whole chapter.

4.1 Structure of the System

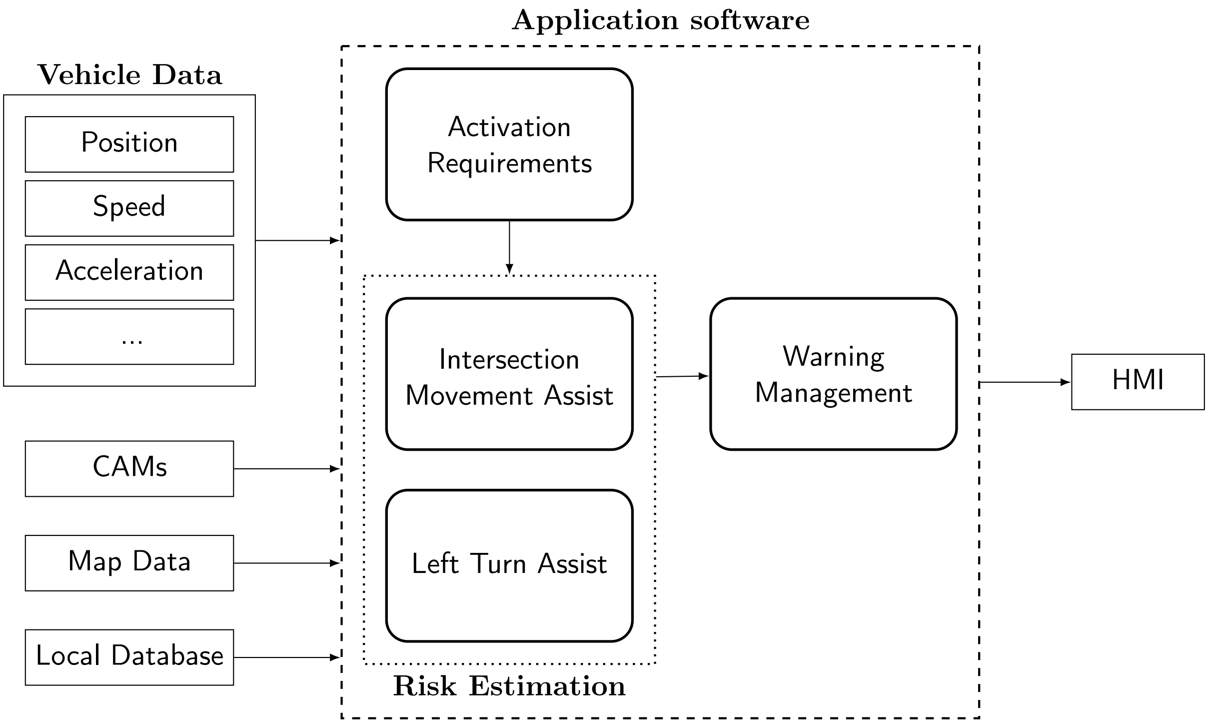


Figure 4.1: Structure of the system.

An overview of the application structure is showed in Figure 4.1. This structure is composed by the following elements:

- **Application Software**: this software runs in a central processor. It processes all incoming variables and informs the HMI-module whenever an intervention is required. Performing the following tasks is responsibility of this software:

    1. Assessment of Activation Requirements: those are the requirements explained in section 3.4.1. This task aims to recognize non-dangerous situations as early as possible in order to save computational time and resources.

    2. Risk Estimation: if all activation requirements are met, the probability of a collision is computed using an adequate calculation method, i.e. IMA for crossing-traffic and LTA for oncoming-traffic. A further explanation of this process is given in section 4.4.

    3. Warning Management: the computed probability is then used for deciding if either a warning, a notification or no information at all is to be given to the end-user.

- **Vehicle Data**: this corresponds to the information explained in section 2.3 which is associated to the current state of the Ego-vehicle and used for the risk estimation.

- **CAMs**: every one of these messages contains information corresponding to the current state of another V2X-capable vehicle. This information is used for determining if the Other-vehicle poses a threat for the Ego-vehicle and for estimating the risk of a collision.

- **Map Data**: the road characteristics needed by the application, as listed in section 2.4, are extracted from this database. This information can also be located in the Local Database.

- **Local Database**: this contains vehicle-related constants, fixed road parameters and other settings for customizing the application. The use of this database allows the deployment in different kind of vehicles without having to modify the source code of the application.

- **HMI**: this refers to the interface with the end-user, which is capable of issuing information at two different levels: notification-level and warning-level. The application software has the responsibility of deciding which level of information is to be used when interacting with the end-user.

4.2 Means for the Implementation

The following subsections describe the resources used for the implementation of the algorithm. The first subsection corresponds to the implementation in a simulation environment while the second one covers the software and hardware implementation on prototype vehicles. Table 4.1 gives a summary of the main features used for the implementation in those environments.

Feature	**Simulation**	**Prototype vehicles**
Framework	CARLA Simulator	Robot Operating System (ROS)
Programming language	Python and C++	Python and C++
Configuration	Input arguments	Parameters in launch-file
Human-Machine Interface	Debug functions	Monitor (over HDMI)
Exchange of information	Function calls	Topics and services
Vehicle data	Virtual sensors	As defined in section 2.3
Vehicle operation	PID controller	Test subjects
Event trigger	Delay-based	Delay-based (via rospy.Rate)

Table 4.1: Features and their implementation in different environments.

4.2.1 Simulation

The present work focuses on the development, evaluation and optimization of a safety-critical application which should intervene in dangerous road situations in order to avoid them. And because of the repeated reproduction of such situations in the real world being hard to accomplish, and also accompanied by dangers for the vehicle drivers, the decision of using a simulation environment for the development and refinement of the algorithms is taken. This also offers the benefit of being able to iteratively test several approaches and configurations, without the dangers of personal and material damage.

The name of the chosen simulation environment is CARLA. This is a three-dimensional, open-source simulator which was primarily conceived for the development and validation of autonomous driving systems [44]. CARLA is implemented as an open-source layer over Unreal Engine 4 (UE4) and posses a server-client architecture[1]. The server has the task of running the simulation and rendering the scene while the client, implemented in Python, is responsible for the exchange of information between agents and the server via sockets. Different commands are sent from the client side to the server in order to operate the agents and to receive information about the current state of the simulation, e.g. sensor readings.

[1] This architecture also allows for great scalability thanks to multiple clients being able to communicate with the server simultaneously.

For the implementation and optimization of the different algorithms, the most recent version of CARLA available at the time of this writing is used, i.e. release 0.9.9. In order to prepare the simulation environment for the execution of the algorithms, a script named "scenario_builder.py" is written using the Python API provided by CARLA. This script is in charge of handling every scene by performing the following steps:

1. **Spawning the vehicles**: the vehicles are positioned near to an intersection, the algorithm to be used is selected according to the input arguments when executing the script and a series of sensors are attached to every vehicle in order to gather the information needed. Among those sensors we have:
    - a GNSS Sensor for gathering the geographical position of the vehicle. This is a conversion of the vehicle location inside the simulation into geographical coordinates.
    - an IMU Sensor for measuring the acceleration and orientation of the vehicle.
    - a Collision Detector for detecting when the vehicles collide with each other.
2. **Operating the vehicles**: the vehicles are operated by a PID controller for navigating through the simulation towards a given destination. This destination and the initial position of the vehicles determine the maneuvers to be performed.
3. **Evaluating the outcome**: every simulation run is evaluated based on the type of scene (dangerous or non-dangerous), the presence or absence of a warning and if a collision took place or not. This is further explained in chapter 5.
4. **Destroying the agents**: all spawned vehicles and sensors are removed from the simulation, giving way to the next scene.

4.2.2 Prototype Vehicles

In order to validate the results obtained through the simulation and to make one further step towards the integration of the algorithms in actual vehicles, the prototype vehicles are equipped with the same application software as the simulated vehicles. The groundwork of those prototype vehicles was conceived within the scope of the Connected Motorcycle Consortium (CMC) for evaluating the potential of different V2X-applications on passenger cars and motorcycles.

The following subsections explain the inner structure of the hardware and software contained in the two prototypes used in this work, as well as the operation of the application software.

System Hardware

The system hardware carries the name of Portable V2X-Setup (PVS) and serves as the medium for encompassing the different tangible and abstract components needed for the implementation of V2X-applications. A diagram of the physical modules contained in the PVS is shown in Figure 4.2 and accompanied by a description of those modules below.

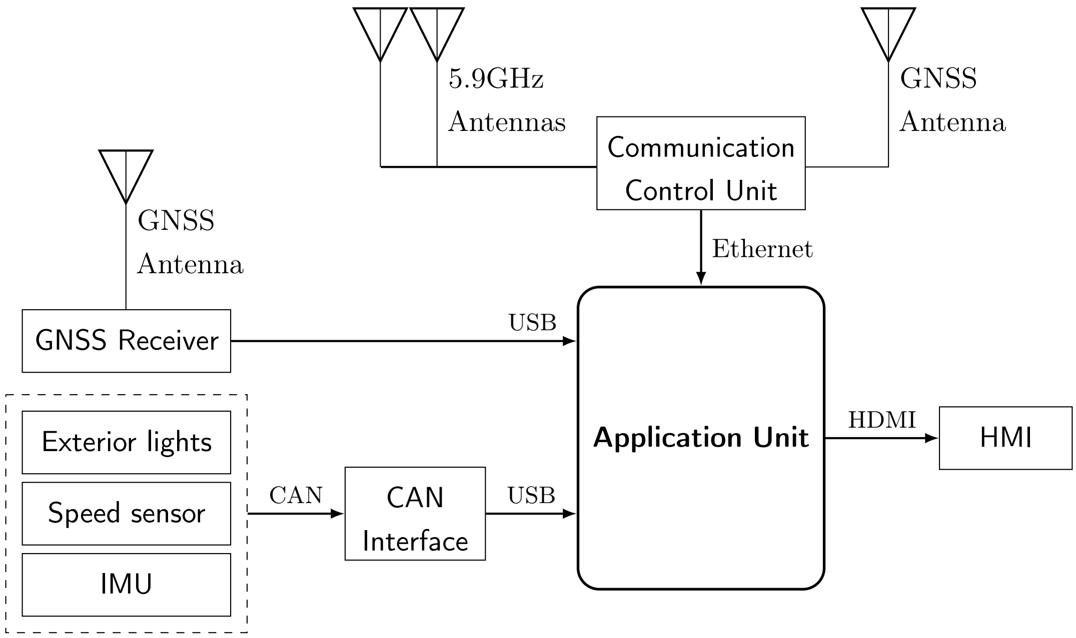


Figure 4.2: System hardware on prototype vehicles.

- **Application Unit**: this is the central processing unit of the system, constituted by an `UDOO x86 II Ultra` running Ubuntu 18.04 and containing the application software.

- **Communication Control Unit**: this unit is responsible for the communication over the PC5 interface, i.e. the interface enabling direct communication between road entities without having to use the cellular network. It contains a development board comprising an `Autotalks CRATON2` chip, which is used exclusively as a transceiver for the exchange of information between vehicles[2].

- **GNSS Receiver**: composed by an `u-blox M8U UDR` module, this module is responsible for receiving and processing satellite signals in order to compute the current geographical position. Additionally, it uses sensor measurements for providing continuous navigation in areas with poor signal reception.

- **Inertial Measurement Unit**: because of it containing its own IMU, the `u-blox M8U UDR` is also being used for measuring the acceleration, angular rate and orientation of the vehicle body.

- **HMI**: this is constituted by a monitor connected to the Application Unit over HDMI. The HMIs for the motorcycle and car prototype are shown in Figure 4.3 and 4.4 respectively.

- **CAN-Interface**: the `USB-to-CAN V2` module acts as the interface between the system and the vehicle-CAN. The obtained information corresponds to the output of the speed sensor and the state of the blinkers.

[2] Some functionalities of this development board, such as GNSS positioning and CAN-interface, are being performed by other modules.

Figure 4.3: Cockpit of motorcycle prototype. The basis for this prototype is a BMW R 1200 RS. Located above the instrumental panel is an additional display for the notifications and warnings given by the V2X-application. Image source: [45].

Figure 4.4: Cockpit of car prototype. The basis for this prototype is constituted by a BMW X5 xDrive40e. The display in the center is being used for the notifications and warnings given by the V2X-application. Image source: [46].

System Software

The system software running on the Application Unit is implemented using ROS (Robot Operating System). This is an open-source framework backed by a large scientific community and providing libraries and tools for the development of robot applications. ROS enables a flexible and scalable development of applications by allowing the modularization of software elements while at the same time taking care of the communication between them.

Every program inside the ROS environment is called a node. Several nodes work together for performing different tasks and interact with each other over two communication models:

- **Publisher/Subscriber model**: this model targets a many-to-many, one-way transport of information between nodes. This exchange of information is based on the use of topics, which are named buses for transmitting or receiving information. A node acts as a publisher when it uses a certain topic for transmitting information, and as a subscriber when it listens to a topic for receiving information coming from other nodes.

- **Request/Response model**: this model uses so called services for establishing an on-demand, client-server communication between nodes. This interaction is based on the use of a pair of messages, one for the client-request and one for the server-response[3].

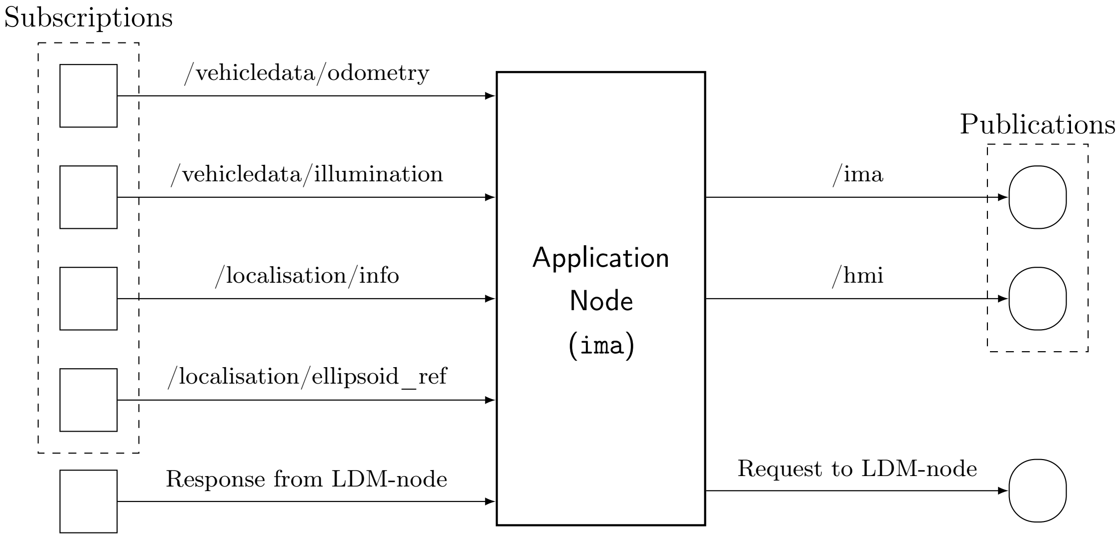


Figure 4.5: Interaction between the application node and other ROS-nodes. The elements on the left correspond to incoming information and the elements on the right correspond to outgoing information. The interaction with the LDM-node does not make use of the publish/subscribe model, but uses service calls instead. LTA is treated here as an use case of IMA and is therefore included in the IMA-node.

[3] Another way of passing data under a request/response model are the so called *Actions*. In contrast to services, these do not enter into standby-mode until a response is received and are therefore used for initiating long-term behaviors. This type of communication is not being used for this implementation.

The IMA application operates inside of this framework in the form of a single node, which is executed periodically every 100 ms. On every execution, this nodes issues a service-request for obtaining the V2X-messages received during the last 100 ms, it then proceeds to process every one of them and publishes on the topic `/ima`, thus signalizing liveliness and depicting the current situation. This node also publishes on the topic `/hmi` when a relevant situation is detected and information is to be given to the end-user. The nodes required by the application and their respective topics are showed in Figure 4.5 and described below.

- **its_ldm**: this node is responsible for the administration of a Local Dynamic Map (LDM) containing timestamped and geographically-referenced information received from other road entities. Service calls (client-requests) are used for interacting with this node in an on-demand manner.

- **vehicledata**: this node collects vehicle information by listening to the CAN-Bus over the CAN-Interface and publishes on the following topics:
  /vehicledata/odometry — containing speed and acceleration.
  /vehicledata/ilumination — containing the state of the blinkers.

- **localisation**: this node collects information from the GNSS receiver, the IMU and the speed sensor. It then applies filters for improving the gathered position and publishes on the following topics:
  /localisation/info — containing the vehicle's heading.
  /localisation/ellipsoid_ref — containing the improved geographical position.

- **hmi**: in order to manage the transmission of information to the end-user, this node listens to and processes the messages published on the following topic:
  /hmi — messages on this topic mainly contain:
  (a) the action to be performed, i.e. activation or revocation of information,
  (b) the name of the image and audio files to be used,
  (c) a priority level for resolving concurrency[4].

Asynchronous callback functions in the `ima` node are in charge of receiving and storing the information on the topics published by `vehicledata` and `localisation`, thus ensuring that this data is available by the time it is needed by the application.

[4] This situation takes place when two or more V2X-applications intend to present information to the end-user at approximately the same time.

Map Data

The solution considered for gathering the map data required by some of the approaches is the use of OpenStreetMap (OSM), a free database of worldwide geographic data. The interaction between the OSM-database and the application software is handled by the Overpass API[5]. This interface is used for obtaining custom selected parts of the OSM-database and was conceived for the consumption of data only, enabling it to have a better performance than APIs capable of both reading from and writing to the OSM-database. This API can access the OSM-database either online as a web-service, or offline after storing a copy of the OSM-world or portions of it.

The process for extracting map data out of the OSM-database is described in Algorithm 4.1. This algorithm, to be executed inside the `ima` node before calling the function for risk estimation, returns the following intersection-related properties, as described in section 2.4:

$$I = (pos, tf, lanes_{ego}, lanes_{other}, road_{ego}, road_{other}, prio_{ego}) \tag{4.1}$$

Where pos corresponds to the geographical coordinates of the intersection, $lanes_{ego}$ and $lanes_{other}$ to the number of lanes in the respective roads, $road_{ego}$ and $road_{other}$ to the type of these roads and $prio_{ego}$ to the Ego-vehicle having right-of-way or not.

Algorithm 4.1 OSM data extraction.

Input: Positions pos_{ego} and pos_{other}; headings h_{ego} and h_{other}.
Output: Intersection properties $I = (pos, tf, lanes_{ego}, lanes_{other}, road_{ego}, road_{other}, prio_{ego})$.

```
w_ego,other ← road(pos_ego,other)                                    ▷ Road names
if w_ego ∩ w_other exists then
    foreach Node n ∈ {w_ego ∩ w_other} do                            ▷ Common node(s)
        b_ego,other ← bearing(pos, pos_ego,other, h_ego,other)       ▷ Rel. bearing w.r.t. intersection
        if abs(b_ego) ≤ 60 and abs(b_other) ≤ 60 then                ▷ Intersection ahead of vehicles
            pos ← coordinates(n)
            tf ← true if related_way(n) ⊃ traffic_signal else false
            lanes_ego,other ← lanes(w_ego,other)
            road_ego,other ← road_type(w_ego,other)
            prio_ego ← true if w_other ⊃ give_way else false
            return I                                                 ▷ Intersection properties
        end if
    end for
end if
return NaN                                                           ▷ No common intersection found
```

This algorithm also forms part of the activation requirements (see section 3.4.1), because of the returned value NaN indicating that the vehicles are not approaching the same intersection.

[5] This API is formerly known as OSM Server-Side-Scripting (OSM3S).

4.3 Parameters and Variables across Approaches

The following subsections comprehend the most important parameters and variables needed by the different approaches in order to compute the risk of a collision.

4.3.1 Geodesic Distance

The distance between two geographical points, i.e. the geodesic distance, is the shortest distance on the surface of the earth connecting those two points. The geographical coordinates of the vehicles are then used for computing the distance between them. This computation is carried out by a function contained in the Python library *geopy*, which implements the method given in [47] for those purposes.

Additionally to the distance between vehicles, the following two distances are also considered by some of the approaches:

- **Distance-to-Collision (DTC)**: this corresponds to the distance between a vehicle and the Point-of-Collision. For the ETSI-based approach, this is calculated by solving a triangle equation for which two angles and one side are known. For the ETSI-based approach having access to map data (ETSI-Map), this distance is obtained by applying the same formula used for the distance between vehicles, with the difference that the arguments are the position of a vehicle and the coordinates of the intersection. A two-dimensional offset given by the relative position of the vehicles is then added or subtracted from this distance.

- **Distance-to-Intersection (DTI)**: this corresponds to the distance between a vehicle and the point at which the intersection begins[6]. The SAE-based approach applies the same formula for determining the distance between vehicles, with the difference of having the position of a vehicle and the center of the intersection as arguments, and then subtracts a two-dimensional offset from this result.

Both the distance-to-collision and the distance-to-intersection could also be computed using the map database. This distance is then obtained by adding the single distances between way-points from the position of the vehicle to the position of the node corresponding to the Point-of-Collision or begin of the intersection. Another option, not employed in this implementation, would be to transform the vehicle geographical coordinates into Cartesian coordinates and then compute the euclidean distance between the resulting points.

4.3.2 Time-to-Collision

The Time-to-Collision (TTC) is a key parameter used across all four approaches. It gives information about the criticality of the situation by indicating in how much time the vehicles are going to be inside the area of conflict, i.e. inside the intersection for the SAE-based approach and inside the Point-of-Collision for the ETSI-based approaches.

[6] The begin of an intersection is defined as point after which two lanes intersect each other.

In order for the algorithm to consider the situation as critical, one of the conditions to be fulfilled is the computed TTC being under a threshold TTC_{min}, as explained in section 3.8. And although not always carrying the same name, the concept of the TTC is used by all presented approaches in the following ways:

- For ETSI-based approaches, the TTC refers to the time required for the center of the vehicles to arrive at the Point-of-Collision.

- For the SAE-based approach, the task of the TTC is performed by the Time-to-Intersection (TTI), which indicates the time required for the vehicles to enter the intersection.

- For the Ghost-vehicles approach, the TTC corresponds to the time for the prediction of the vehicle's position, i.e. the ghost vehicle is to be located TTC seconds ahead of the actual vehicle.

It is worth noting that the optimal values for this parameter may vary for different approaches. For example, when using the same parameter value, the SAE-based approach would tend to warn earlier than the ETSI-based approaches, because of the former calculating the time to enter the intersection and the latter the time for arriving at the Point-of-Collision, which is inside the intersection.

Computation of the TTC

The easiest way of calculating a Time-to-Collision t consist of making use of the distance to the area of conflict d and the speed of the vehicle v, as expressed in equation 4.2.

$$t = \frac{d}{v} \tag{4.2}$$

The issue with using equation 4.2 is that it does not take a very important physical factor into consideration, the acceleration. The calculation of a TTC without the involvement of the acceleration would lead to a very short-sighted value, which doesn't express the state of the vehicle in the near future.

In order to calculate a TTC while taking the acceleration of the vehicle into consideration, we part from the equation for the final velocity v_f in function of the initial velocity v_i, acceleration a and Time-to-Collision t. We then rearrange it for obtaining t, as shown in equation 4.4.

$$v_f = v_i + a\ t \tag{4.3}$$

$$t = \frac{v_f - v_i}{a} \tag{4.4}$$

Then we take an equation relating distance, acceleration and time, and substitute t for equation 4.4, resulting in the following:

$$d = \frac{1}{2}\ a\ t^2 = \frac{1}{2}\ a \left(\frac{v_f - v_i}{a}\right)^2 = \frac{1}{2}\ \frac{v_f^2 - v_i^2}{a} \tag{4.5}$$

We can then rearrange the rightmost part of equation 4.5 for obtaining v_f:

$$v_f = \sqrt{2\ a\ d + v_i^2} \tag{4.6}$$

The final step is to substitute v_f in 4.4 for equation 4.6, resulting in:

$$t = \frac{\sqrt{2\ a\ d + v_i^2} - v_i}{a} \tag{4.7}$$

Equation 4.7 is then expressed in terms of the distance to the area of conflict, the current acceleration and the initial speed of the vehicle, i.e. the speed at the beginning of the calculation. This equation can be used for calculating either TTC or TTI simply by specifying the distance to be used: DTC or DTI.

Optimal Threshold for TTC

In order for a situation being considered as critical, the value for TTC obtained through equation 4.7 must be below a threshold TTC_{min}. This threshold should be set neither too high nor too low. If it is set too high, the algorithm would react too-early and could classify non-dangerous situations as critical, and if it is set too low, the end-user could not have any time left for avoiding the collision. In order to obtain an adequate value for TTC_{min}, different approaches are considered for its approximation:

- **Approximation based on ETSI's formula**: This analytical approximation, introduced in section 3.5.2, aims for the vehicle to completely stop before arriving at the conflict area. The TTC_{min} corresponding to each driving speed can be estimated using equation 4.4 while having $v_f = 0$. Figure 4.6 shows the resulting values for TTC_{min} when having a deceleration of $7\,\mathrm{m/s^2}$, a MDRT of 1.2 s and a MLT of 0.2 s.

- **Approximation based on accident data**: This approximation is made using accident data from the GIDAS-database and also aims for a complete stop from the Ego-vehicle before entering the conflict area [48]. For its calculation, a deceleration of $7\,\mathrm{m/s^2}$ and a reaction time of $1s$ are used. Figure 4.6 shows the obtained values for TTC_{min}.

- **Approximation of a fixed value**: Another possibility is the estimation of a fixed value for TTC_{min} over the whole speed range. This approximation method is further treated in section 4.5.

Figure 4.6 shows a comparison between the mentioned methods for the approximation of TTC_{min}. In this Figure, the closer the value is to the x-axis, the later a situation is seen as critical. For giving an idea of how frequent the different values for TTC_{min} are used, the bar graph shows a distribution corresponding to the percentage of collision accidents according to the initial speed of the vehicles [48].

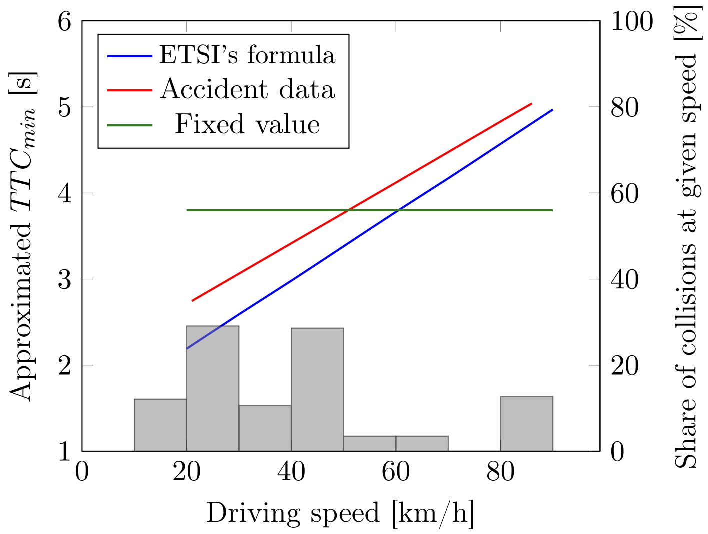
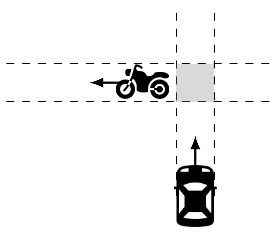


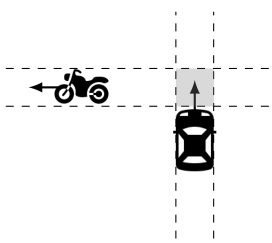
Figure 4.6: Approximation of TTC_{min} based on different methods. The x-axis represents the driving speed of the Ego-vehicle, the y-axis on the left corresponds to the threshold value calculated using the indicated methods and the y-axis on the right shows the percentage of registered collision accidents having the speed on the x-axis as the initial speed, i.e. the driving speed before trying to avoid the collision. The highest shares on this bar graph correspond to three different groups of locations: 30 km/h zones, 50 km/h zones and rural areas.

4.3.3 Encroachment Time

The Encroachment time (t_{enc}) is another key parameter for estimating the criticality of the situation involving the Ego-vehicle. It is based on the Post Encroachment Time (PET), which represents the time difference between a vehicle leaving an area of conflict, i.e. area of encroachment, and an adversary vehicle entering this same area [11]. Figure 4.7 illustrates this concept by showing the state of the vehicles at the time-points delimiting the PET.



(a) time-point t_1



(b) time-point t_2

Figure 4.7: Time-points delimiting the Post Encroachment Time. The motorcycle is leaving the area of conflict (gray area) at time t_1, while the car is arriving at this same area at time t_2.

Having those time-points, the PET is calculated as follows:

$$PET = t_2 - t_1 \tag{4.8}$$

In this work, the Encroachment time is defined as the absolute time difference between the center of the vehicles arriving at the area of conflict. This is expressed as follows:

$$t_{enc} = \begin{cases} abs(TTC_{ego} - TTC_{other}) & \text{for ETSI-based approaches,} \\ abs(TTI_{ego} - TTI_{other}) & \text{for SAE-based approach.} \end{cases} \tag{4.9}$$

The closer t_{enc} gets to 0, the higher the probability of a collision. A t_{enc} value of 0 then implies that a collision is 100 % certain, as both vehicles would be at the area of conflict at exactly the same time.

Similar to the TTC_{min} previously explained, a t_{enc_min} is used as the lowest absolute value of Encroachment time allowed before issuing a warning.

4.3.4 Driver Maneuver

In order to choose the most relevant parameters for the risk estimation and also identify the possible Point-of-Collisions with other road users, it is necessary to know the intention of the drivers, i.e. which maneuver they are about to perform at the intersection.

Assessing the intention of other drivers exclusively through V2X-communication poses a challenge for the system. This is due to the limited information contained in the transmitted CAMs and the possible absence of significant information such as the state of the blinkers and the path history[7].

The maneuver corresponding to the Ego-vehicle, on the other hand, can be estimated through different methods such as:

- Lane context: the intention of a driver can be estimated by knowing on which lane the vehicle is located and which are the maneuvers allowed for the lane in question.
- Intersection characteristics: the topological layout of an intersection delivers information about the available maneuvers for the drivers.
- Driving behavior: information such as the path history, the current vehicle orientation and the state of the blinkers can be used for estimating which maneuver the driver is going to perform next.

There are different approaches for maneuver estimation, which use some the methods mentioned above, or a combination of them [8, 49]. In this work, the state of the blinkers is used as the indicator for the maneuver estimation, because of this information directly reflecting the intention of the driver.

[7] These information is contained in the Low-Frequency Container of CAMs, whose implementation is not mandatory and transmission rate is set to 1Hz, i.e. only 10 % of the transmitted messages would contain this information.

4.3.5 Bearing Angle

The bearing angle θ between two geographical coordinates is defined as the angle measured in the clockwise direction from the north line going vertically through one of the coordinates and to the line segment connecting both coordinates, as shown in Figure 4.8.

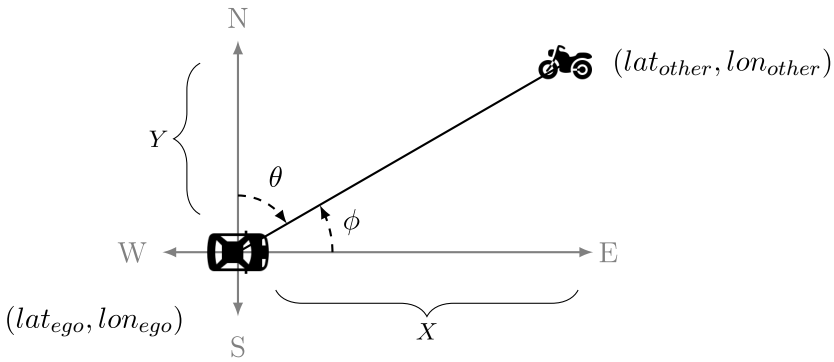


Figure 4.8: Bearing angle between two geographical coordinates. Angle θ corresponds to the bearing with respect to the north, while angle ϕ represents the bearing angle with respect to the front of the Ego-vehicle. X and Y are the horizontal and vertical distances between the vehicles over the surface of the earth.

Given the latitude and longitude of both vehicles, the bearing angle with respect to the true north can be calculated by applying the following formula:

$$\theta = \arctan \frac{X}{Y} \tag{4.10}$$

$$\theta = \arctan \frac{\sin(lon_{other} - lon_{ego}) \ \cos(lat_{other})}{\cos(lat_{ego}) \ \sin(lat_{other}) - \sin(lat_{ego}) \ \cos(lat_{other}) \ \cos(lon_{other} - lon_{ego})} \tag{4.11}$$

As the resulting bearing angle is expressed in terms of the true north instead of in terms of the Ego-vehicle, a relative bearing ϕ is calculated by subtracting the heading of the Ego-vehicle from the obtained bearing angle and normalizing the result in order to obtain values between $-180°$ and $180°$.

$$\phi = \begin{cases} \theta - heading_{ego} & \text{if } -180° \leq (\theta - heading_{ego}) \leq 180°, \\ \theta - heading_{ego} + 360° & \text{if } (\theta - heading_{ego}) < -180°, \\ \theta - heading_{ego} - 360° & \text{if } (\theta - heading_{ego}) > 180°. \end{cases} \tag{4.12}$$

Through this normalization, negative values of ϕ indicate that the Other-vehicle is located on the left side of the Ego-vehicle, while positive values indicate the opposite.

4.3.6 Orientation and Heading

In this work, those two concepts are treated differently and defined as follows:

- **Orientation** is the compass direction in which the front of the vehicle is pointed[8]. This direction is related to the spatial position of the vehicle and can be measured through a magnetic compass.
- **Heading** is the compass direction in which the vehicle travels. This direction is thus movement-related and often obtained through the combination of magnetic sensors, acceleration sensors and GNSS-receivers. In the field of navigation, this magnitude often receives the name of *course*.

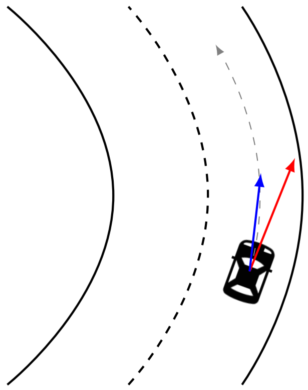


Figure 4.9: Comparison between orientation and heading. The vehicle shown in the diagram is driving along the curve in the road, following the gray, dashed line. The red line corresponds to the orientation of the vehicle, while the blue line corresponds to its heading or direction of travel.

Because of the orientation of the vehicles not always corresponding to the driving direction, the heading is the magnitude which, together with the bearing angle, is employed by the different algorithms in order to determine:

(a) if the vehicles are approaching or getting away from each other.
(b) whether the current situation corresponds to a cross-traffic or oncoming-traffic scenario.

4.4 Implementation of Approaches for Collision-Detection

This section covers the implementation of the four different collision-detection approaches presented in section 3.5. For every presented approach, the idea behind its implementation, the parameters for its adjustment and the corresponding algorithm are explained in detail. Every approach considers the use cases corresponding to both IMA and LTA, as well as the information levels MAI and MAW, i.e. the Ego-vehicle is able to receive both notifications and warnings.

[8] This direction is measured in degrees, the true north corresponding to 0°.

4.4.1 SAE-based

For this approach, previously introduced in subsection 3.5.1, up to three relevant zones are built around the center of the intersection in front of the Ego-vehicle. These zones are shown in Figure 4.10 and described below:

- The relevant zone on the left is present on every scenario.
- The relevant zone on the right is present when no intention of turning right is detected, as there would be no crossing of trajectories with a vehicle coming from the right.
- The relevant zone above the intersection is present when the left blinker is activated in order to take oncoming-traffic into consideration.

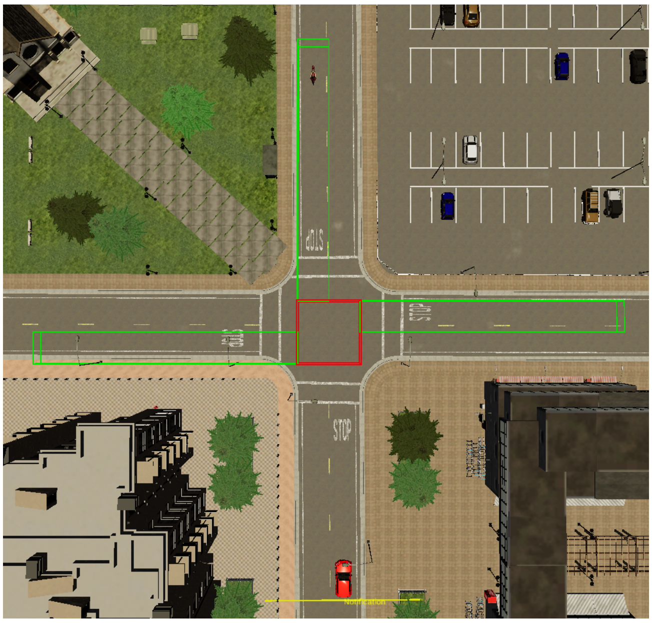


Figure 4.10: Relevant zones of SAE-based approach. The green boundary boxes represent the relevant zones within the simulation environment. Inside the simulation, issued notifications and warnings are presented graphically through a yellow or red line, their position indicating the time of reception. Here, the Ego-vehicle receives a notification because of the Other-vehicle having entered one of the relevant zones.

As soon as the Other-vehicle enters a relevant zone, the Ego-vehicle receives a notification and the application proceeds to compute the need of a warning. For this computation, the DTI and TTI of both vehicles are determined and used for the calculation of the Encroachment time t_{enc}.

In order to issue a warning, the following conditions must hold:

$$TTI_{ego} < TTI_{min} \tag{4.13}$$

$$t_{enc} < t_{enc\_min} \tag{4.14}$$

They verify that both vehicles are arriving at the intersection at approximately the same time and that this is about to happen in the near future.

Parameters

- Minimum Time-to-Intersection (TTI_{min}) [s]: this indicates the lowest threshold that the TTI can have before considering the situation as critical.
- Minimum encroachment time (t_{enc_min}) [s]: this parameter is used for assessing the probability of both vehicles entering the intersection at the same time.
- Length of relevant zones (len_{rz}) [m]: this length can be either a fixed value based on the speed limit for the current road or a dynamical value calculated using the speed of the vehicles. The width of each relevant zone is selected for covering all adjacent lanes towards the intersection. The speed-dependent estimation of the length is made taking into consideration that time gaps under 4 s are normally not used for crossing an intersection [50].

Algorithm

The algorithm 4.2 uses the previously described method in order to issue either a warning, a notification or no information at all. The input arguments are the following:

- Vehicle data corresponding to the Ego-vehicle, which includes its position, speed, heading, acceleration and state of the blinkers:

$$VD = (pos_{ego}, v_{ego}, h_{ego}, a_{ego}, blk_{ego}) \tag{4.15}$$

- Information about the Other-vehicle contained in the received CAM:

$$CD = (pos_{other}, v_{other}, h_{other}, a_{other}, length_{other}, width_{other}) \tag{4.16}$$

- Intersection properties I, as the output of algorithm 4.1.

4.4.2 ETSI-based

This approach, introduced in section 3.5.2, estimates a Point-of-Collision using the position and driving direction of both vehicles. This Point-of-Collision is obtained by solving a triangle equation, for which two angles and one side are known, and is then used for the calculation of the TTC and Encroachment time. Figure 4.11 shows the implementation in the simulation and illustrates the triangle-solving process.

Algorithm 4.2 Algorithm for the SAE-based approach.

Input: Vehicle data VD; received CAM data CD, Intersection properties I.
Output: Time t_{enc} and *HMI-Message* $\in$ (*Warning, Notification, No-information*).
Parameters: TTI_{min}, t_{enc_min} and len_{rz}.

$RZ \leftarrow relevant\_zones(pos, lanes_{other}, road_{other}, len_{rz}, blk_{ego})$
foreach Zone $z \in RZ$ **do**
  if $z \supset pos_{other}$ **then**
    $DTI_{ego,other} \leftarrow geo\_distance(pos, pos_{ego,other})$
    $TTI_{ego,other} \leftarrow t(DTI_{ego,other}, v_{ego,other}, a_{ego,other})$ ▷ Using equation 4.7
    $t_{enc} \leftarrow abs(TTI_{ego} - TTI_{other})$
    if $(t_{enc} < t_{enc\_min})$ **and** $(TTI_{ego} < TTI_{min})$ **then**
      return t_{enc}, *Warning* ▷ Critical situation
    end if
    return t_{enc}, *Notification* ▷ Relevant situation
  end if
end for
return $t_{enc} = \infty$, *No-information* ▷ No relevant or critical situation

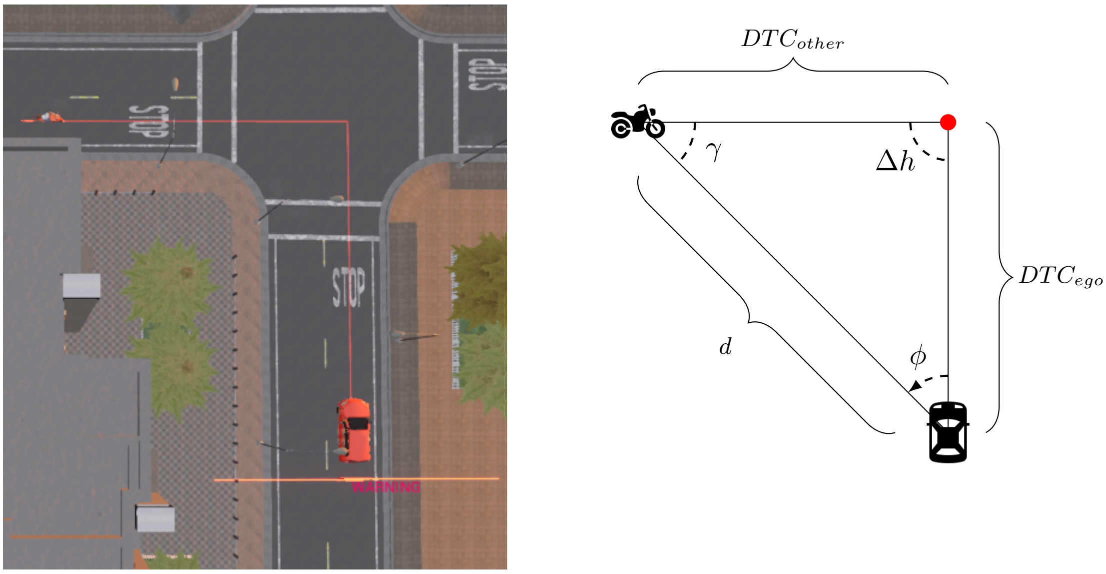


Figure 4.11: Triangle solving for ETSI-based approach. The image on the left corresponds to the implementation of this approach in the simulation environment while the image on the right shows the variables being used by the algorithm. The red line below the Ego-vehicle indicates that it just received a warning.

From this triangle we already know side d which represents the distance between the vehicles and two angles ϕ and Δh corresponding to the relative bearing and to the difference in heading.

The third angle can be found by applying the following formula:

$$\gamma = 180^\circ - \phi - \Delta h \tag{4.17}$$

Next, we can compute the Distance-to-Collisions by using the following two equations:

$$DTC_{other} = d\,\frac{\sin\phi}{\sin\Delta h} \tag{4.18}$$

$$DTC_{ego} = d\,\frac{\sin\gamma}{\sin\Delta h} \tag{4.19}$$

Then we can obtain the value of TTC_{ego} and TTC_{other} through equation 4.7 using the corresponding Distance-to-Collision, vehicle speed and acceleration.

Those values are then used for calculating t_{enc} and checking if the following conditions hold:

$$TTC_{ego} < TTC_{min} \tag{4.20}$$

$$t_{enc} < t_{enc\_min} \tag{4.21}$$

If both of them hold, a warning must be given, as both vehicles would soon arrive at the Point-of-Collision, at approximately the same time. Exactly the same procedure is used for issuing notifications, with the difference of using a higher value for TTC_{min}.

For covering LTA use cases, a triangular Dynamic Safety Shield is used in addition to the circular one. This is activated when the intention of turning left is detected and attempts to detect oncoming-traffic. Figure 4.12 illustrates this.

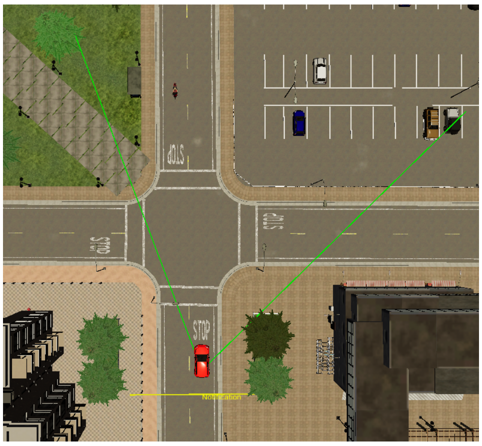


Figure 4.12: Triangular DSS for LTA use case. The green lines represent the boundaries of the triangular DSS. In this case, the Ego-vehicle receives a notification because of the Other-vehicle being located inside of the DSS.

In this case, notifications are issued as soon as the Other-vehicle enters the DSS. Warnings are never given, because of the system being unable to estimate the turning point of the Ego-vehicle, which is mandatory for the estimation of a Point-of-Collision.

Parameters

- Minimum Time-to-Collision (TTC_{min}) [s]: this indicates the lowest threshold that the TTC can have before considering the situation as critical.
- Minimum encroachment time (t_{enc_min}) [s]: this parameter is used for assessing the probability of both vehicles entering the intersection at the same time.
- Radius of circular Dynamic Safety Shield (DSS_{radius}) [m]: this parameter adjusts the overall coverage of the algorithm, being part of the activation requirements (see section 3.4.1).
- Degrees of aperture for triangular Dynamic Safety Shield (DSS_{right} and DSS_{left}) [°]: only applying to LTA use cases, this parameter determines the area to be monitored in front of the Ego-vehicle.

Algorithm

Algorithm 4.3 describes the process for risk computation based on the ETSI guidelines. It takes as arguments the vehicle data and received CAMs, as defined in subsection 4.4.1.

Algorithm 4.3 Algorithm for the ETSI-based approach.

Input: Vehicle data VD; received CAM data CD.
Output: Time t_{enc} and *HMI-Message* ∈ (*Warning, Notification, No-information*).
Parameters: TTC_{min}, t_{enc_min}, DSS_{radius}, DSS_{right} and DSS_{left}.

```
DDS ← circular_shield(pos_ego, DSS_radius)
if DDS ⊃ pos_other then
    if blk_ego ⊃ turn_left then
        t_DDS ← triangular_shield(pos_ego, DSS_right, DDS_left)
        if t_DDS ⊃ pos_other then
            return t_enc = ∞, Notification                 ▷ Relevant LTA scenario
        end if
    end if
    d ← geo_distance(pos_ego, pos_other)
    rb ← rel_bearing(pos_ego, pos_other, h_ego)             ▷ Using equation 4.12
    DTC_ego,other ← distance(d, rb, h_ego, h_other)         ▷ Using equations 4.19 and 4.18
    TTC_ego,other ← t(DTC_ego,other, v_ego,other, a_ego,other)   ▷ Using equation 4.7
    t_enc ← abs(TTC_ego − TTC_other)
    if (t_enc < t_enc_min) and (TTC_ego < TTC_min) then
        return t_enc, Warning                               ▷ Critical situation
    end if
    return t_enc, Notification                              ▷ Relevant situation
end if
return t_enc = ∞, No-information                            ▷ No relevant or critical situation
```

4.4.3 ETSI-based with Map Data

Section 3.5.3 introduced this method and explained the major difference with respect to the ETSI-based approach, i.e. the use of map data for estimating the Point-of-Collision. By using the properties of the intersection and the position, orientation and maneuvers of the vehicles, a Point-of-Collision is estimated. For an intersection between two roads having one lane in each direction, one of twenty[9] possible Point-of-Collisions is selected. This is shown in Figure 4.13.

Figure 4.13: Point-of-Collision estimation based on the use of map data. The Point-of-Collision estimated using map data is represented by the red dot inside of the intersection. In the left picture, both vehicles intend to drive straight across the intersection while, in the picture on the right, the car is intending to perform a right turn.

Map data is also useful for avoiding considering places others than intersections, e.g. road curves and under-crossings. Additionally, it enables warnings for LTA scenarios, as the system can estimate a Point-of-Collision based on where the driver is intending to turn left.

Parameters

- Minimum Time-to-Collision (TTC_{min}) [s]: this indicates the lowest threshold that the TTC can have before considering the situation as critical.
- Minimum encroachment time (t_{enc_min}) [s]: this parameter is used for assessing the probability of both vehicles entering the intersection at the same time.
- Radius of circular Dynamic Safety Shield (DSS_{radius}) [m]: this parameter adjusts the overall coverage of the algorithm, being part of the activation requirements (see section 3.4.1).

[9] These twenty Point-of-Collisions are a sub-set of the 32 conflict points found on a 4-legged intersection. Not considered are the four conflict points nearest to the center of the intersection as well as the eight diverging conflict points, i.e. when leaving a lane by turning left or right.

Algorithm

The following algorithm corresponds to the modification of the previous approach in order to exploit the availability of map data for estimating the Point-of-Collision. It takes as arguments the vehicle data, received CAMs and the properties of the intersection as defined in subsection 4.4.1.

Algorithm 4.4 Algorithm for the ETSI-based approach using map data (ETSI-Map).

Input: Vehicle data VD; received CAM data CD, Intersection properties I.
Output: Time t_{enc} and *HMI-Message* $\in$ (*Warning, Notification, No-information*).
Parameters: TTC_{min}, t_{enc_min}, DSS_{radius}, DSS_{right} and DSS_{left}.

```
DDS ← circular_shield(pos_ego, DSS_radius)
if DDS ⊃ pos_other then
    poc ← collision_point(I, pos_ego, pos_other, h_ego, h_other, blk_ego)
    DTC_ego,other ← geo_distance(poc, pos_ego,other)
    TTC_ego,other ← t(DTC_ego,other, v_ego,other, a_ego,other)    ▷ Using equation 4.7
    t_enc ← abs(TTC_ego − TTC_other)
    if (t_enc < t_enc_min) and (TTC_ego < TTC_min) then
        return t_enc, Warning                                    ▷ Critical situation
    end if
    return t_enc, Notification                                   ▷ Relevant situation
end if
return t_enc = ∞, No-information                                 ▷ No relevant or critical situation
```

4.4.4 Ghost Vehicles

As explained in section 3.5.4, this approach is based on predicting the future position of the vehicles according to the current vehicle state and assuming a constant driving behavior. The ghost vehicles are also able to perform the maneuver intended by the driver through the use of map data and the state of the blinkers. An offset for the position of the ghost vehicles is determined by computing the distance to be travelled in time t_{pred} and performing the corresponding maneuver, in case the distance to be travelled is higher than the distance to the intersection. The distance to be travelled is calculated using the following formula:

$$d_{ghost} = v\ t_{pred} + \frac{1}{2}\ a\ (t_{pred})^2 \tag{4.22}$$

For this implementation, two classes of ghost vehicles are generated in front of every participant, one for the warnings and another for the notifications. Figure 4.14 shows the implementation of these ghost vehicles in a cross-traffic scenario.

Figure 4.14: Ghost vehicles generated in the simulation. Two ghost vehicles are generated for every vehicle. The red ones are used for giving warnings, while the yellow ones are predicted further away and used for giving notifications.

After the corresponding ghost vehicles are generated, the overlap of the ghost vehicles is determined by measuring the distance between the ghost vehicles of the Ego-vehicle and the ones of the Other-vehicle. A warning is given if the distance between two ghost vehicles is below a threshold d_{min}.

$$distance(ghost_{ego},\ ghost_{other}) < d_{min} \tag{4.23}$$

By considering the second class of ghost vehicles and using a higher value for d_{min}, condition 4.23 can be also used for giving notifications.

Parameters

- Minimum distance between ghost vehicles (d_{min}) [m]: this corresponds to the distance between the center of the vehicles involved.
- Prediction time for warnings (t_{pred}) [s]: this parameter is equivalent to the concept of TTC explained before and indicates the amount of seconds between the actual vehicles and their ghost vehicles for issuing warnings.
- Prediction time for notifications (t_{pred_not}) [s]: being similar to t_{pred}, this parameter indicates the amount of seconds between the actual vehicles and their ghost vehicles for issuing notifications.
- Size of ghost vehicle (g_{length} and g_{width}) [m]: these parameters describe the width and length of the ghost vehicles, which may be differ from the actual dimensions of the real vehicle.

Algorithm

The following algorithm shows the procedure for the . It takes as arguments the vehicle data, received CAMs and the properties of the intersection as defined in subsection 4.4.1.This algorithm always returns the distance between the ghost vehicles, which represent the position of the actual vehicles t_{pred} seconds into the future.

Algorithm 4.5 Algorithm for the approach based on ghost vehicles.

Input: Vehicle data VD; received CAM data CD, Intersection properties I.
Output: Distances d_{ghost} and *HMI-Message* $\in$ (*Warning, Notification, No-information*).
Parameters: d_{min}, t_{pred}, t_{pred_not}, g_{length} and g_{width}.

$d_{ghost\_ego} \leftarrow pos\_offset(v_{ego}, a_{ego}, t_{pred})$ ▷ Using equation 4.22
$ghost_{ego} \leftarrow gen\_ghost(pos_{ego}, d_{ghost\_ego}, g_{length}, g_{width}, blk_{ego})$

$d_{ghost\_other} \leftarrow pos\_offset(v_{other}, a_{other}, t_{pred})$
$ghost_{other} \leftarrow gen\_ghost(pos_{other}, d_{ghost\_other}, length_{other}, width_{other})$

$d_{ghost} \leftarrow distance(ghost_{ego}, ghost_{other})$
if $d_{ghost} < d_{min}$ **then**
    return d_{ghost}, *Warning* ▷ Critical situation
end if

$d_{ghost\_not\_ego} \leftarrow pos\_offset(v_{ego}, a_{ego}, t_{pred\_not})$
$ghost\_not_{ego} \leftarrow gen\_ghost(pos_{ego}, d_{ghost\_not\_ego}, g_{length}, g_{width}, blk_{ego})$

$d_{ghost\_not\_other} \leftarrow pos\_offset(v_{other}, a_{other}, t_{pred\_not})$
$ghost\_notother \leftarrow gen\_ghost(pos_{other}, d_{ghost\_not\_other}, length_{other}, width_{other})$

$d_{ghost\_not} \leftarrow distance(ghost\_not_{ego}, ghost\_not_{other})$
if $d_{ghost\_not} < d_{min}$ **then**
    return d_{ghost}, *Notification* ▷ Relevant situation
end if
return d_{ghost}, *No-information* ▷ No relevant or critical situation

4.5 Optimization of Parameters Based on Simulation

Calculation Method for Minimum TTC

The method selected in this work for the calculation of TTC_{min} is the one based on the formula given by ETSI. This method is considered to transparently cover all important factors involved in the avoidance of collisions. It is also able to take different contexts into consideration, i.e. driving speed and braking capabilities of the vehicle.

The use of a fixed value for TTC_{min} is not further considered because it lacks the ability of covering scenarios in all speed ranges. A low, fixed value of TTC_{min} would properly suit low-speed scenarios, but it would tend to give warnings too late when confronted with higher-speed scenarios. And a high, fixed value of TTC_{min} could cover lower-speed scenarios but would tend to give warning too early, which often leads to a decrease in acceptance [13]. This effect is illustrated in Figure 4.15.

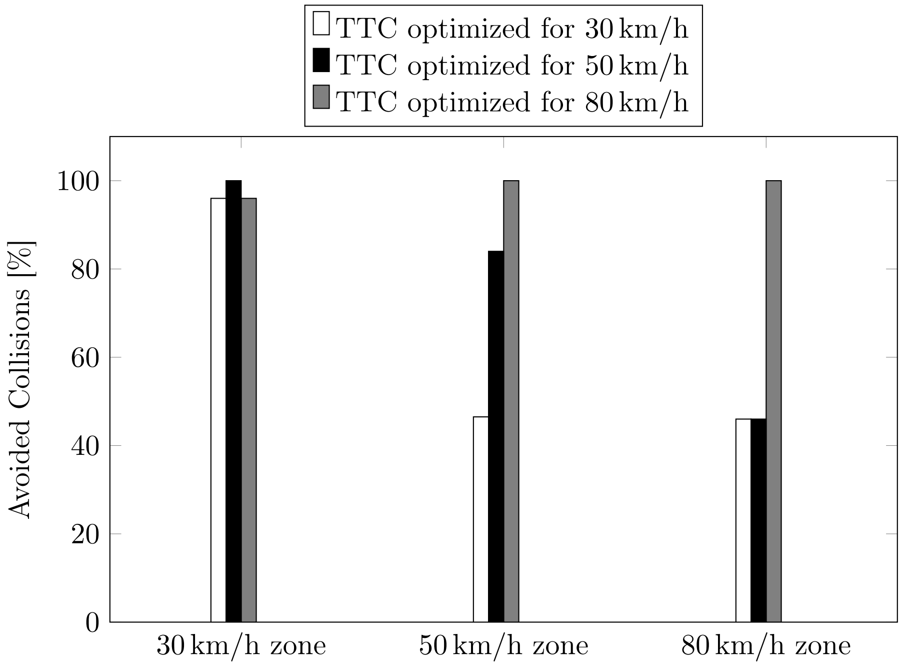


Figure 4.15: Relative performance of a fixed TTC_{min} on different speed zones. This evaluation is made across different scenarios using the ETSI-based approach and a t_{enc_min} equal to 0.1 s. The percentages of avoided collisions are normalized to the highest share of avoided accidents for a given speed zone.

Minimum Encroachment Time

The value for the minimum encroachment time (t_{enc_min}) and distance (d_{min}) has a direct impact on the proportion of dangerous and non-dangerous situations being correctly identified by the algorithms.

Figure 4.16 shows the effect of different t_{enc_min} and d_{min} values over the performance of the different algorithms. These results are obtained through the simulation and illustrate the trade-off between true negatives and true positives when this parameter is changed. These two indicators, explained in section 5.3, refer to the amount of dangerous situations for which a warning is given (true positives) and the amount of non-dangerous situations for which no warning is given (true negatives).

The results on Figure 4.16 indicate that a higher value of t_{enc_min} or d_{min} makes the algorithms more sensitive, thus increasing the share of true positives. This higher t_{enc_min} or d_{min} value also has the effect of decreasing the percentage of true negatives, because of the algorithms falsely classifying non-dangerous situations as dangerous situations.

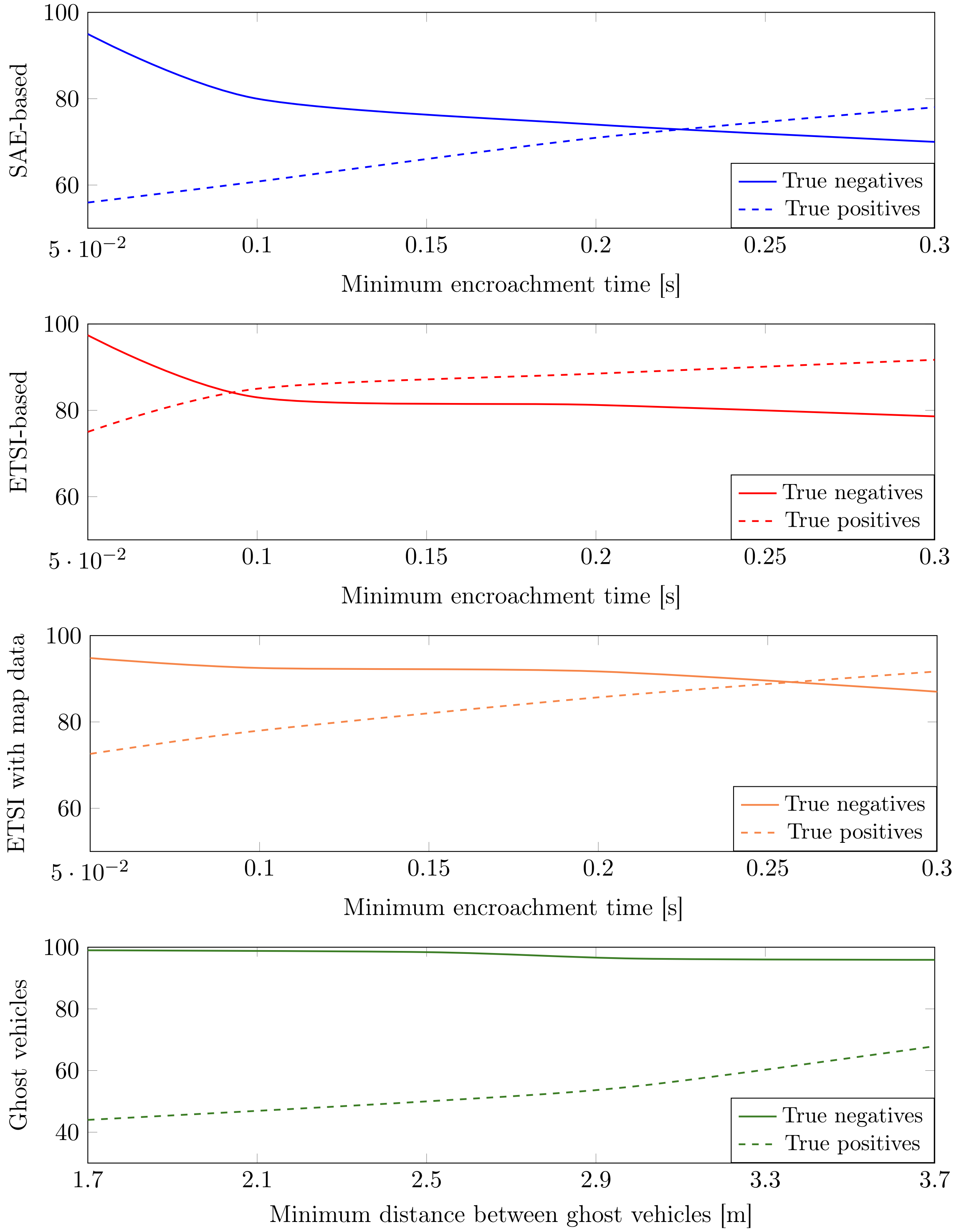


Figure 4.16: True negatives and true positives in relation to minimum encroachment time and minimum distance. Solid lines represent the true negatives while dashed lines represent true positives for the warnings given by the different algorithms.

4.6 Discussion

This chapter deals with the structure and implementation of the system in a simulation environment as well as in prototype vehicles. The simulation environment acts as a flexible framework for the development and evaluation of the different approaches, while the vehicle prototypes serve as a validation tool for the results obtained.

The CARLA Simulator is the chosen simulation environment for this work. Its set of sensors is used in order to simulate the sensor system on a real vehicle, i.e. GNSS receiver, speed sensor, IMU, etc. A script named *scenario_builder.py* is developed in order to manage the simulation environment and to prepare the vehicles and scenarios to be evaluated. This script also serves as the information channel between the vehicles and is in charge of assessing the result for every single scenario.

For the prototype vehicles, a hardware including all necessary sensors and modules is incorporated into them. This hardware is capable of running ROS, in which the application IMA is contained in form of a process node and interacts with the rest of the system via topics and services.

All relevant variables and parameters handled by the different algorithms are also introduced. The most relevant for estimating the criticality of the road situation being the TTC and Encroachment time. The former indicates the time left for taking a collision-avoidance action and its lowest threshold is calculated by taking the braking capabilities of the vehicle and the driver reaction time into consideration. The latter is used for estimating the probability of a collision and its lowest threshold manages the trade-off between the rate of true positives and true negatives given by the algorithms.

The implementation of the algorithms corresponding to the four different approaches is also described. The calculation methods for both notifications and warnings are included, which are implemented in order to test the potential of issuing information at the levels MAI and MAW. These algorithms are designed for flexibility, and their parameters can be used for adapting them to different driver profiles and vehicle classes. Two important assumptions are also made during the development of these algorithms. The first one is the proper usage of the blinkers by the end-user having the application on-board and the second one is a constant acceleration rate (positive or negative) while approaching an intersection.

5 Evaluation

This section offers a theoretical and empirical analysis on the performance of the implementation of the four different approaches. It begins with generation of the data set used for the evaluation of the algorithms and the selected models for recreating the driver reaction upon reception of a warning. It continues explaining the different indicators used for assessing the performance of the approaches and then presents the results of simulation and driving trials, thus providing an insight into the capabilities and potential of every approach.

5.1 Data Set

The data set created for this evaluation can be divided into categories, which are composed by several scenarios and intend to test distinct properties of the different approaches.

The classification used for the reconstruction of accidents in Germany is taken as a reference for the construction of the scenarios for this evaluation [51]. This classification distinguishes between 7 different types of accidents, each one describing the properties of the road, the kind of users involved (e.g. vehicle, pedestrian, cyclist), their initial position with respect to each other and the maneuver they perform (e.g. turning left, turning right, going straight).

The two relevant types of accidents for this work are Type 2 and Type 3, which describe accidents between at least two vehicles taking place at intersections:

- **Accidents while turning (Type 2)**: this type stands for accidents between two vehicles which drive in the same or in opposite directions. The oncoming-traffic scenarios belong to this category and represent the uses cases for LTA.
- **Accidents while crossing or merging (Type 3)**: this represents accidents taking place at a crossing or T-junction and in which the vehicles drive in non-parallel directions, thus corresponding to the use cases for IMA.

Some of the accident codes contained in each accident type are group together because of each one of these accident types covering several kinds of situations while only some of these situations are of relevance for the applications IMA and LTA. Table 5.1 shows the accident codes of the different situations. The details of the clustering made for this evaluation can be found in Appendix 7.3.

Scenario	Code	EV	OV
Oncoming-traffic (LTA)	211	↰	↑
	212	↰	↱
	215*	↰	↰
	351	↑	↰
Cross-traffic OV from left (IMA)	301	↑	↑
	302	↰	↑
	303	↱	↑
	306*	↰	↱
	326*	↱	↱
Cross-traffic OV from right (IMA)	321	↑	↑
	322	↰	↑
	323*	↱	↑

Table 5.1: Accident codes used for evaluation. Columns EV and OV corresponds to the action taken by the Ego-vehicle and Other-vehicle, respectively. A star next to a code indicates that this scenario does not lead to a collision under normal driving conditions, e.g. both vehicles turning left in front of each other.

The selected accident codes are used for populating the following categories of scenarios:

- **Collision**: scenarios at which a collision does take place. These scenarios are used for testing the collision-detection properties of every approach, as they enable the estimation of collisions predicted in-time, too late or not at all.

- **No-collision**: scenarios at which a collision could occur but does not occur. Those scenarios test the fine-tuning of the approaches by confronting them with situations which could appear to be critical, but does not end in a collision, thus testing their tendency of giving false warnings.

- **Safe**: scenarios at which no collision can occur. On these scenarios, the trajectories of the vehicles do not intersect each other, which allows for testing the robustness of the approaches when a collision is not likely.

Although the accident codes in the category *Safe* are found only in this category, the scenarios of both *Safe* and *No-collision* are used for the estimation of false warnings.

Generation of Scenarios

Having the starting position and maneuvers of the vehicles, which are prescribed by the accident code, the corresponding scenarios can be generated.

In order to make the scenarios as representative as possible, the speed values for the scenarios are chosen according to the cumulative distribution of speed in zones with different speed limits. According to the bar graph presented in Figure 4.6, the number of relevant zones amounts to three: 30 km/h, 50 km/h and 80 km/h zones[1]. This graph also indicates the amount of scenarios to be generated for every speed zone.

The cumulative functions corresponding to the 30 km/h and 50 km/h speed distributions are derived from [52], while the distribution for 80 km/h is estimated based on the first two. The resulting speed distributions are shown in Figure 5.1.

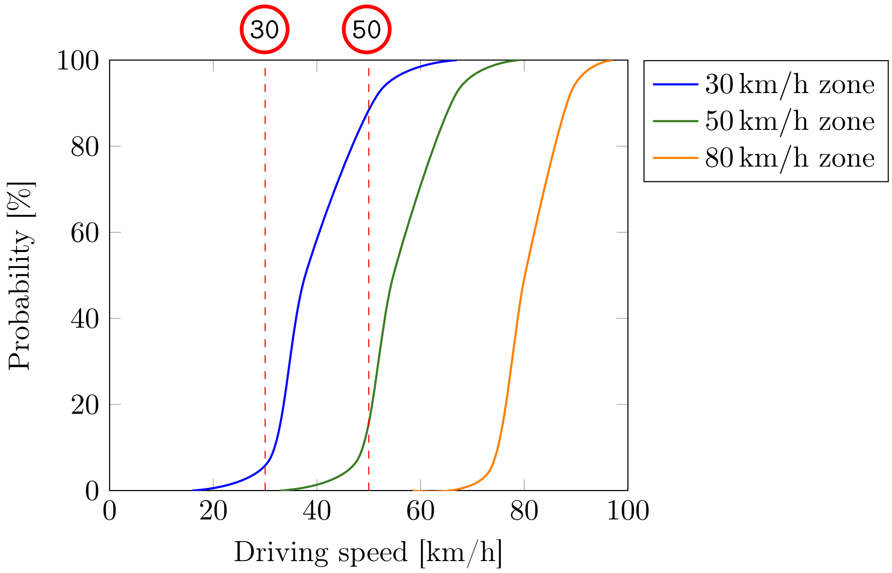


Figure 5.1: Driving speeds in zones with different speed limits. Cumulative distribution function for driving speeds inside of 30 km/h, 50 km/h and 80 km/h zones.

All three distributions rely on the concept of v_{85}-speed, which describes the driving speed for 85 % of the road users. Those were obtained from sensor measurements in different locations and used for generating v_{85}-speed ranges corresponding to the different zones[2].

[1] The 80 km/h zone corresponds to the driving speed at crossings and T-junctions on rural areas.
[2] The v_{85}-speeds for 30 km/h and 50 km/h zones cover the ranges 31-52 km/h and 48-67 km/h respectively [52].

5.2 Driver Model for Warning Reactions

Not all drivers react the same way when confronted with a dangerous road situation, and in order to test the different approaches against different human responses, a driver model for warning reactions is implemented. The human responses in dangerous road situations can be classified according to two different factors:

- **Reaction time**: this is the time a driver needs between receiving an information and taking a certain evasive action, e.g. perform an emergency brake.
- **Breaking intensity**: this refers to how intensively a driver brakes. For this evaluation, and in order to cover different road and tire conditions, intervention of ABS and cohesion coefficients, the maximum possible deceleration[3] (intensity of 100 %) is set to $7\,m/s^2$.

In this work, six different models for simulating driver reactions upon reception of a warning are used. Those were estimated in [54] by combining the three most probable reaction times with two observed braking intensities, a strong one and weak one. The resulting models and their probability of occurrence are presented below:

Model	**1**	**2**	**3**	**4**	**5**	**6**
Reaction time	0.72 s	0.54 s	0.72 s	1.06 s	0.54 s	1.06 s
Braking intensity	100 %	100 %	50 %	100 %	50 %	50 %
Probability of occurrence	0.36	0.27	0.12	0.12	0.09	0.04

Table 5.2: Reaction models according to their probability of occurrence.

5.3 Performance Evaluation

The performance of the different approaches is primarily measured in terms of accidents avoided thanks to a warning, accidents not avoided due to non-detection and the rate of false warnings, i.e. warnings given during not-dangerous situations [55].

In order to evaluate these performances, every scenario is simulated at least twice. A first time without the intervention of the collision-avoidance algorithm, i.e. the algorithm remains disabled during the whole scenario, and a second time having the algorithm enabled. The first run serves the purpose of knowing the outcome of the scenario in advance, the scenario can then be classified into the previously introduced categories *Collision* or *No-collision*.

[3] The deceleration of a vehicle in good technical condition while braking on dry asphalt is in the range of 6.0–$7.85\,m/s^2$, according to [53].

Performance indicators

For the evaluation of the different approaches, the following performance indicators are used:

(a) **Percentage of successfully avoided collisions**: this is the number of collisions that didn't happen in the simulation for which a warning is given and the total number of collisions occurred in the simulation when there is no intervention by the algorithm.
(b) **Percentage of too-late detected collisions**: this is the ratio between the number of collisions occurred in the simulation for which a warning is given and the total number of collisions occurred in the simulation when there is no intervention by the algorithm.
(c) **Percentage of not-detected collisions**: this is the ratio between collisions occurred in the simulation for which no warning is given and the total number of collisions occurred in the simulation when there is no intervention by the algorithm.
(d) **Percentage of true positives**: this is given by the ratio between the number of detected collisions (avoided or not) and the total number of collisions occurred in the simulation when there is no intervention by the algorithm. The number of detected collisions is the sum of (a) and (b).
(e) **Percentage of false positives**: this is defined as the percentage of non-dangerous situations for which a warning is given.
(f) **Percentage of false negatives**: this is equivalent to the percentage of not-detected collisions (c).
(g) **Percentage of true negatives**: this is defined as the complementary of the percentage false positives.

5.4 Simulation Results

This section presents the results obtained through the simulation. In order to determine the strengths and weaknesses of the approaches for different scenarios, multiple speeds are generated stochastically for all accident codes. The total amount of scenarios used for the evaluation is shown in Table 5.3.

Category	Accident codes	Quantity	Total
Collision	8	24	192
No-collision	8	24	192
Safe	4	24	96
			480

Table 5.3: Amount of scenarios for evaluation inside the simulation.

All different approaches are confronted with the same test scenarios, i.e. the same sets of data, and executed using all six models for driver reaction. The results for the different driver models are weighted according to their probability of occurrence and combined in a single statistic.

5.4.1 SAE

The following graphs show the results obtained for the SAE-based approach under the different categories of scenarios. For the parameter t_{enc_min}, a value of 0.1 s is used.

Detection accuracy

Figure 5.2 presents the results for the category *Collision*. For this category, three performance indicators are obtained and associated to the corresponding accident code as well as to the corresponding use cases, i.e. either IMA or LTA.

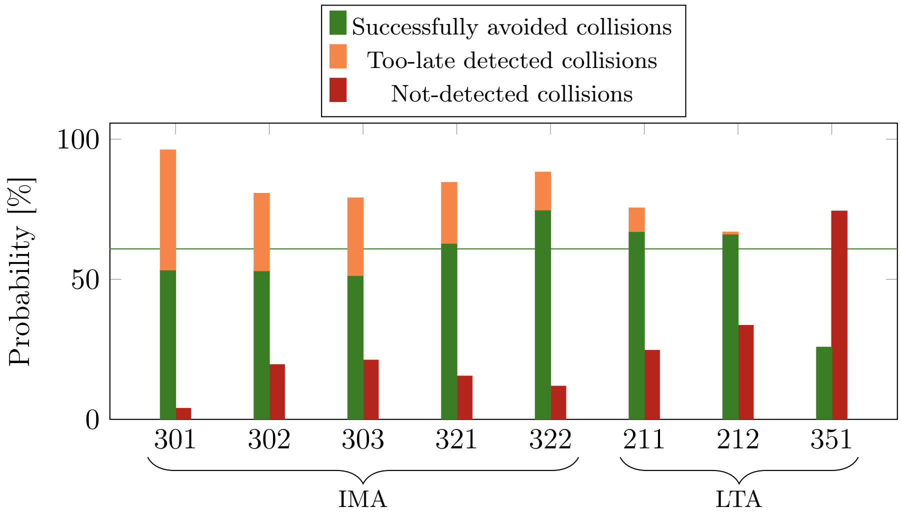


Figure 5.2: Performance of SAE-based approach for *Collision* scenarios. Green line represents average performance when not covering accident code 351.

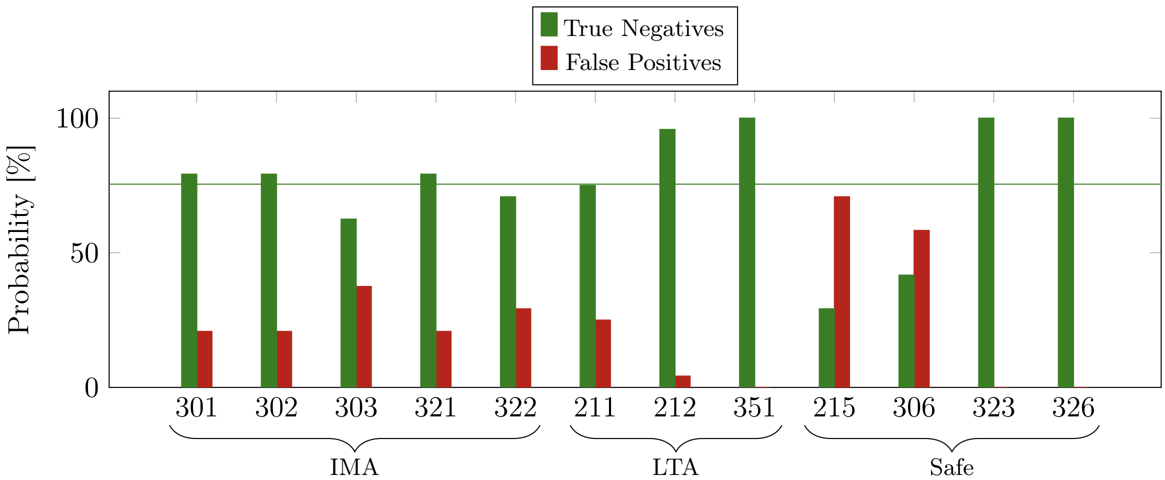


Figure 5.3: Performance of SAE-based approach for *No-collision* and *Safe* scenarios. Green line represents average performance.

Figure 5.3 shows the results for the categories *No-Collision* and *Safe* together, as they are both used for the estimation of true negatives and false positives, i.e. false warnings.

Runtime

Figure 5.4 then illustrates the runtime of the SAE-based algorithm for the three different possible outcomes, i.e. warning, notification and no-information. The runtime for every outcome is expressed in terms of the average time, the time corresponding to the 25^{th} and 75^{th} percentile, as well as the maximum and minimum time needed in order to estimate the risk after having the information about the properties of the intersection.

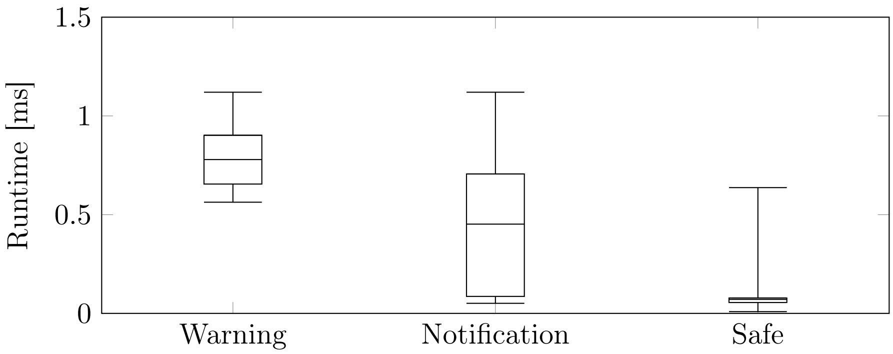


Figure 5.4: Runtime of SAE-based algorithm. Those times are obtained through the simulation environment running on the hardware specified in Appendix 7.5.

Interpretation of results

This approach demonstrated being capable of avoiding over 60 % of the collisions, with only one scenario being well below the average, i.e. 351. This scenario corresponds to the use case "Other-vehicle turning left ahead" and is specially challenging because of the Ego-vehicle not knowing the Other-vehicle's intention of turning left.

The average proportion of false negatives is below 25 %, with only two scenarios showing a high rate of false warnings, i.e. 215 and 306. Those scenarios correspond to the cases in which both vehicles turn left in front of each other and in which the Other-vehicle is approaching the Ego-vehicle from the left side and intends to turn right at the intersection. Just as for the scenario 351, the lack of information about the next maneuver of the Other-vehicle makes these scenarios challenging for the algorithm.

The runtime inside the simulation environment is constantly below the limit of 1 ms. The speed of computations for non-dangerous scenarios being relatively high in comparison to the other two. This is due to the relevant areas delimiting the area of interest and quickly ignoring vehicles outside of those.

5.4.2 ETSI

The simulation results obtained for this approach are presented in the same manner as the ones in the previous section. The value of the parameter t_{enc_min} for this evaluation corresponds to 0.2 s.

Detection accuracy

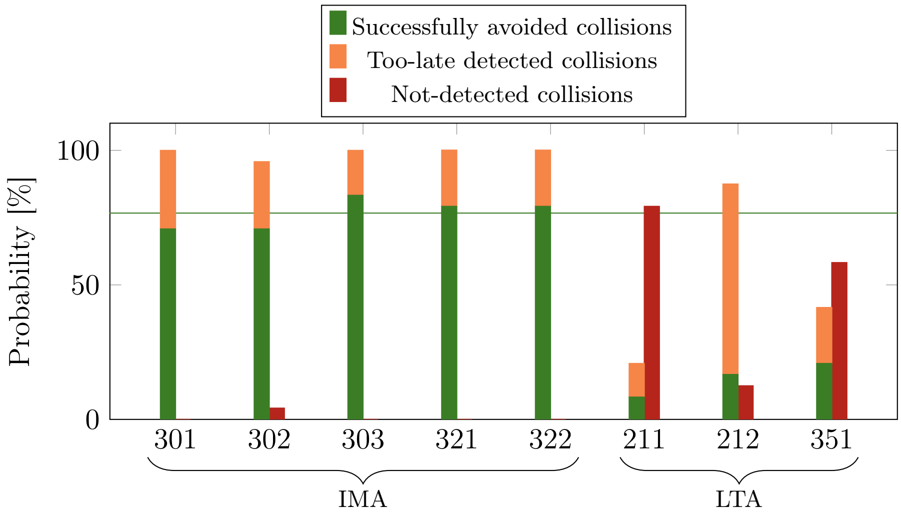


Figure 5.5: Performance of ETSI-based approach for *Collision* scenarios. Green line represents average performance when not covering LTA use cases.

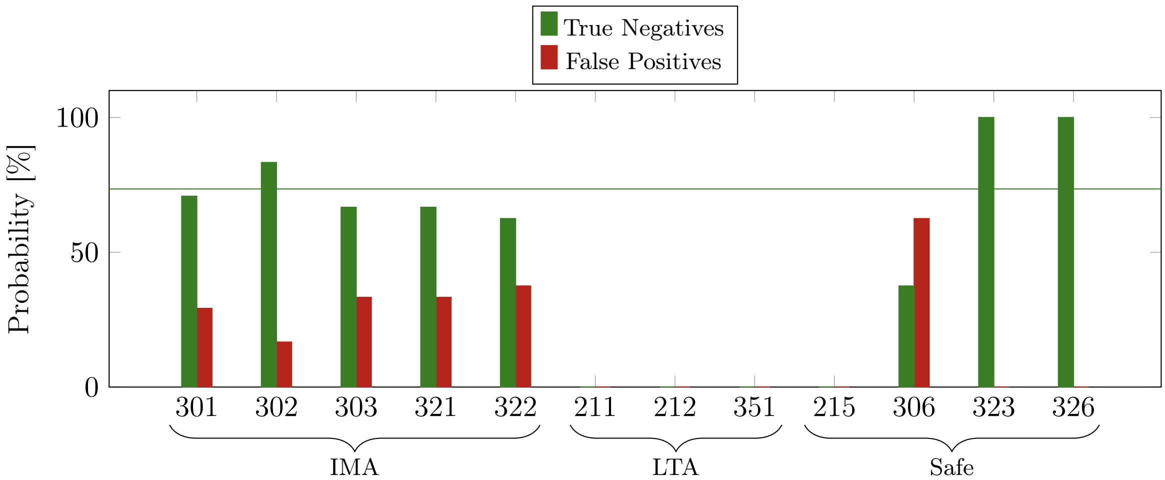


Figure 5.6: Performance of ETSI-based approach for *No-collision* and *Safe* scenarios. Green line represents average performance. The LTA use cases, i.e. 211, 212, 351 and 215, are excluded because of this approach not being able to issue warnings under such circumstances.

Runtime

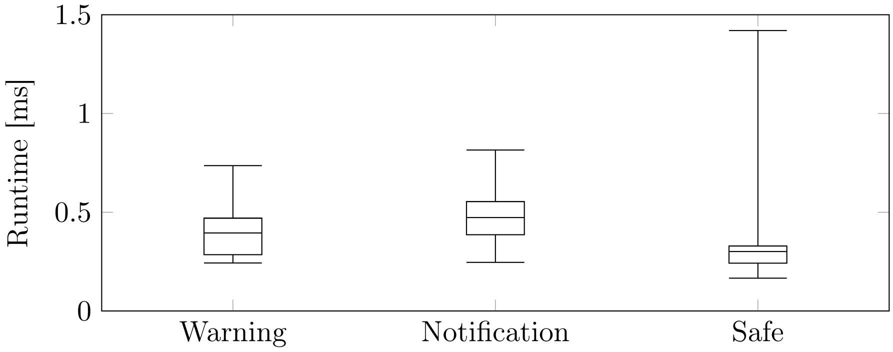


Figure 5.7: Runtime of ETSI-based algorithm. Those times are obtained through the simulation environment running on the hardware specified in Appendix 7.5.

Interpretation of results and special considerations

It is noticed that the detection accuracy of this approach is sufficient for issuing LTA-notifications but not for giving LTA-warnings. This due to the turning point of the Ego-vehicle being unknown to the algorithm. The runtimes for both warnings and notifications are similar, because of the algorithm performing the same computations for both of them. The time for assessing non-dangerous situations varies greatly in some cases due to the need of solving triangle equations before classifying them as non-dangerous.

Figure 5.8: ETSI-based approach estimating a Point-of-Collision out of the road. Because of the lack of map data, the ETSI-based approach does not distinguish between intersections and curves. This could lead to the calculation of unreachable Point-of-Collisions when two vehicles approach a curve.

An important aspect to be considered for this algorithm are the false warnings taking place outside of intersections. The lack of map data could lead to the estimation of a Point-of-Collision located outside of an intersection, which in turn leads to possible false warnings. This is illustrated in Figure 5.8.

5.4.3 ETSI with map data

Below are the result for the ETSI-based approach using map data for estimating the location of the Point-of-Collision. A value of 0.2 s is given to the parameter t_{enc_min}.

Detection accuracy

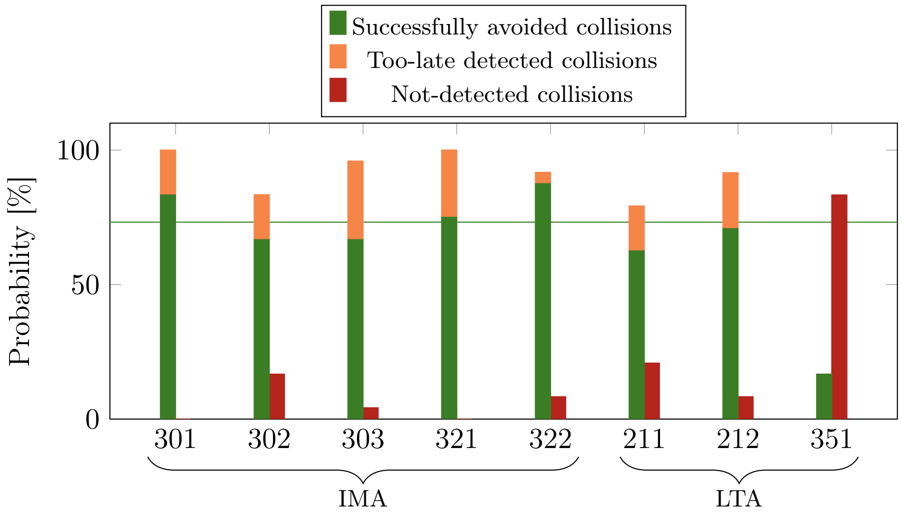


Figure 5.9: Performance of ETSI-Map approach for *Collision* scenarios. Green line represents average performance.

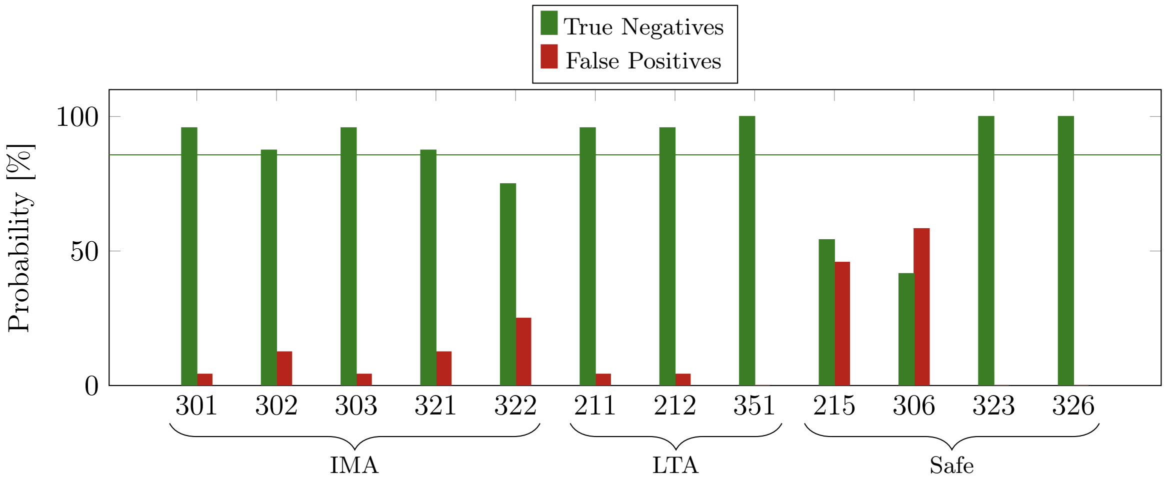


Figure 5.10: Performance of ETSI-Map approach for *No-collision* and *Safe* scenarios. Green line represents average performance.

Runtime

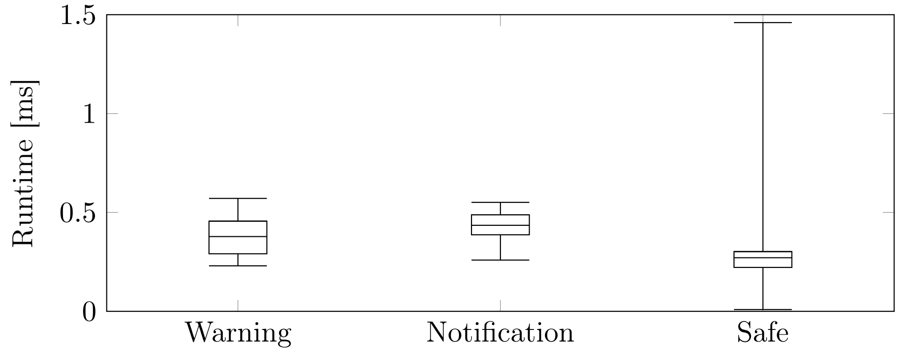


Figure 5.11: Runtime of ETSI-Map algorithm. Those times are obtained through the simulation environment running on the hardware specified in Appendix 7.5.

Interpretation of results

In contrast to its parent approach, this one demonstrates a certain ability to avoid collisions in LTA use cases. This is due to the additional information about the location of the intersection, which is used for estimating the possibles Point-of-Collisions.

This approach, similar to the SAE-based approach, tends to give false warnings for the scenarios 215 and 306, as well as not giving any warnings for the scenario 351. This is largely due to the lack of information about the maneuver the Other-vehicle is about to perform.

The estimated runtimes are similar to the ones of the ETSI-based approach. This is primarily due to both approaches calculating the TTC and evaluating the risk of a collision in the same manner.

5.4.4 Ghost Vehicles

This subsection presents the results for the approach based on the prediction of the vehicle's position, i.e. the generation of ghost vehicles. The algorithm for this approach is evaluated using $d_{min} = 3.7\,\mathrm{m}$.

Detection accuracy

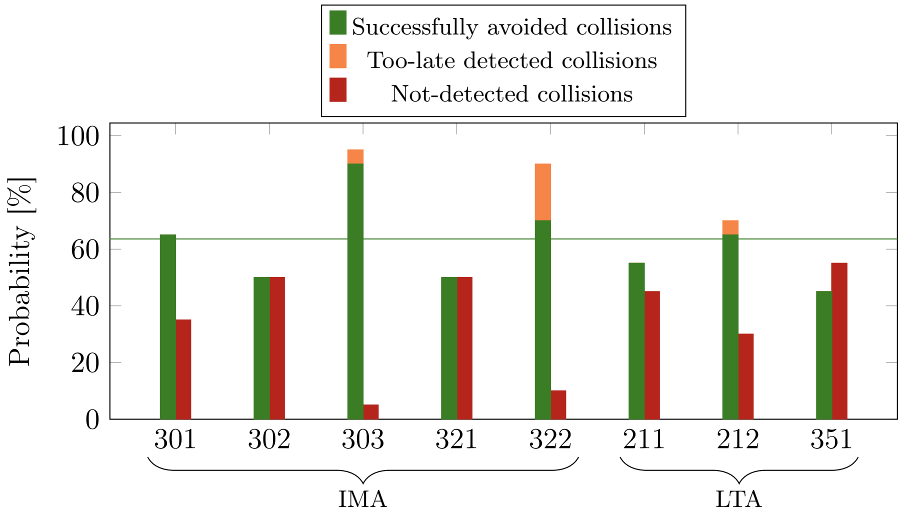


Figure 5.12: Performance of approach based on ghost vehicles for *Collision* scenarios. Green line represents average performance of this algorithm.

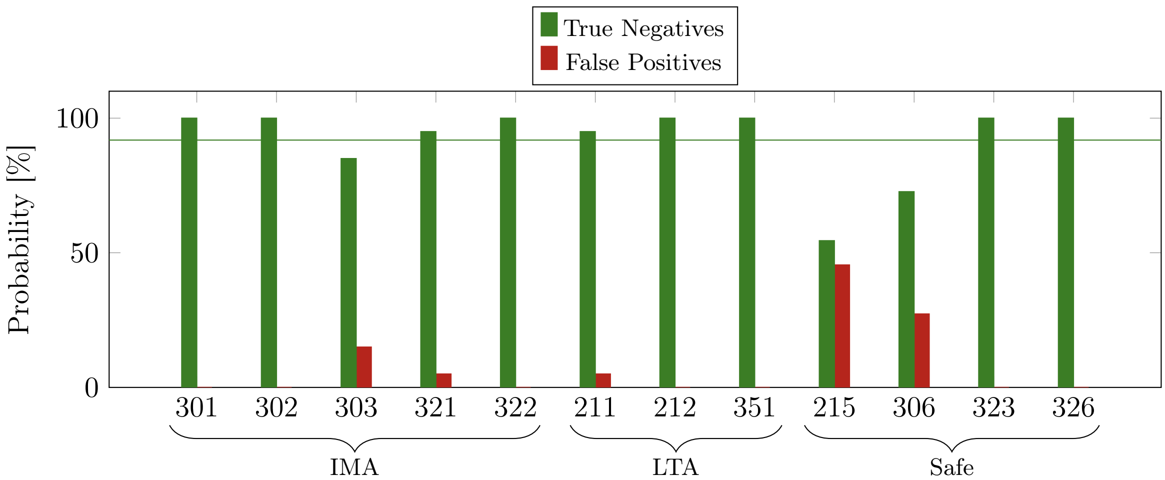


Figure 5.13: Performance of approach based on ghost vehicles for *No-collision* and *Safe* scenarios. Green line represents average performance.

Runtime

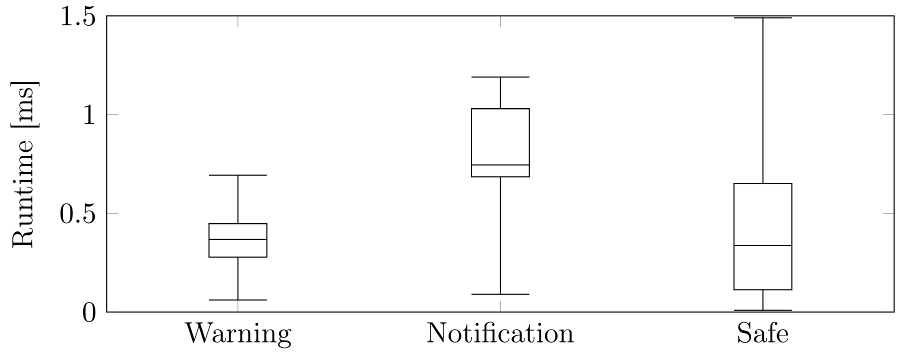


Figure 5.14: Runtime of algorithm based on ghost vehicles. Those are obtained through the simulation running on the hardware specified in Appendix 7.5.

Interpretation of results and special considerations

In contrast to other approaches, only a small amount of collisions are detected too late by this algorithm. This is because of this approach not searching continuously for collisions in the entire trajectory of the vehicle. This property leads to high number of true negatives and a relatively low number of true positives.

The runtime of this algorithm for the issue of notification is always higher than for issuing warnings. This is due to the ghost vehicles for notifications being generated only after no overlap between the ones for warnings is detected.

For the implementation of this approach, special consideration needs to be given to the number of lanes connected to the intersection. When operating at an intersection with multiple lanes for a same direction of travel, a lane-level position accuracy may be needed in order to correctly allocate the ghost vehicle. This means that the algorithm needs to know on which lane the vehicles are located, before predicting their future position.

5.4.5 Comparison across Approaches

Figure 5.15 and 5.16 give an overview of the collision-avoidance potential of every approach, as well as their robustness against false warnings across the different scenarios.

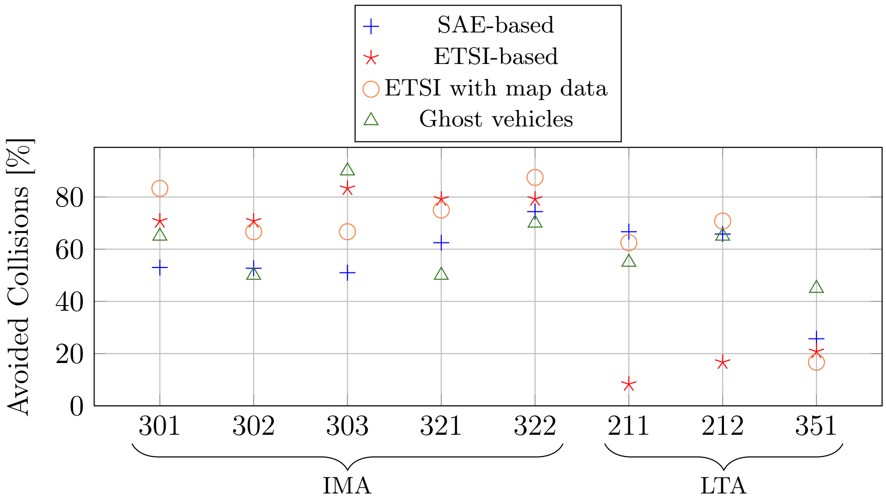


Figure 5.15: Percentage of avoided collisions across different approaches.

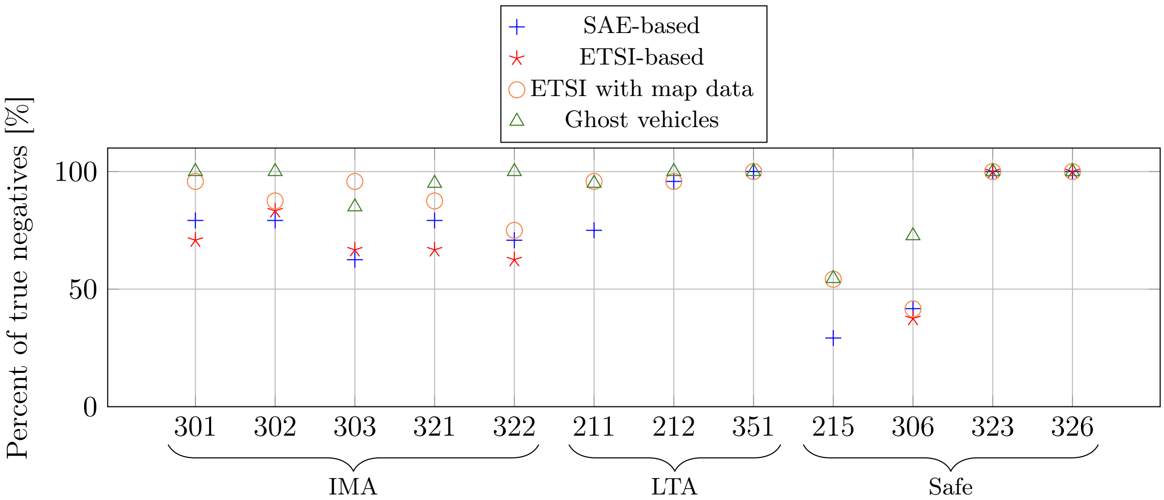


Figure 5.16: Rate of true negatives across different approaches.

5.4.6 Autonomous Interventions

An autonomous intervention is here defined as a collision-avoidance maneuver performed by the vehicle without requiring a human action. A vehicle can intervene at a later point than a driver, because of the MDRT not having to be considered anymore [56]. This allows for the road situation to become clearer before deciding if an intervention is needed, thus being able to reduce the rate of false warnings. This autonomous intervention is implemented in the form of a 7$^{\text{th}}$ driver reaction model having the following properties:

- Reaction time = 0 s
- Brake intensity = 100 %

Figure 5.17 exhibits the potential of autonomous interventions across the different approaches. This potential is based on the amount of collisions being avoided due to a direct intervention of the system, i.e. the system performs a braking maneuver without any input from the end-user.

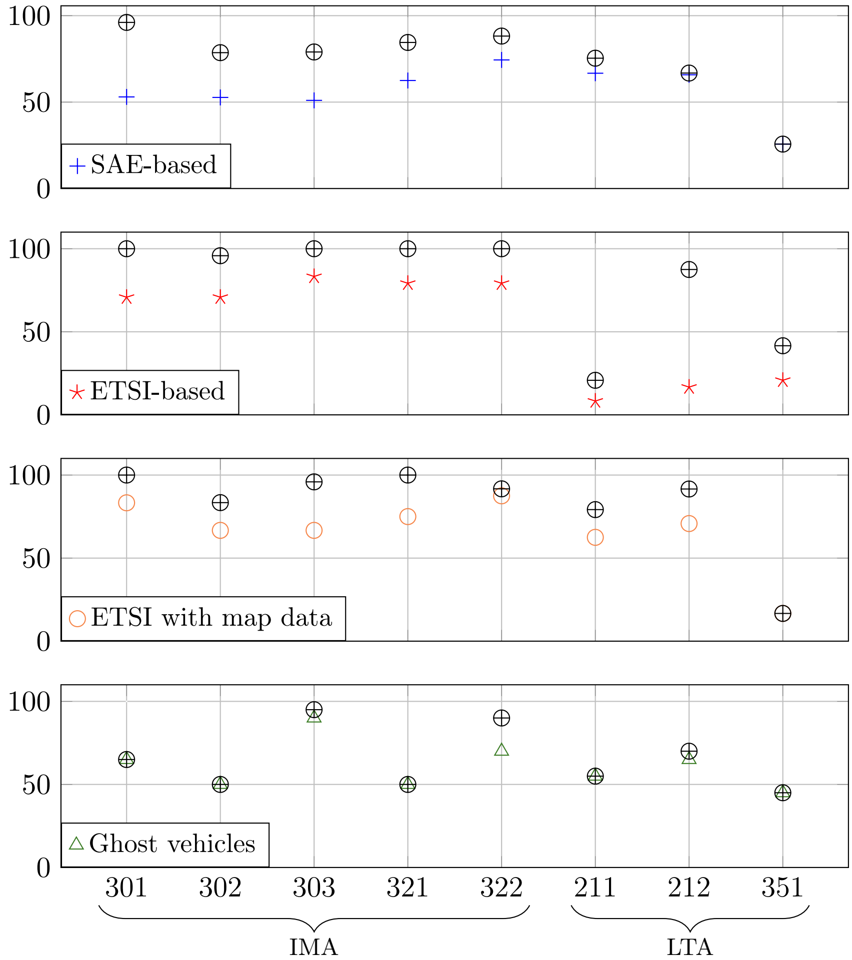


Figure 5.17: Potential of autonomous interventions for increasing safety. The y-axis correspond to the percentage of avoided collisions. The black marks represent the potential increase in safety due to autonomous interventions.

All four approaches benefit differently from such autonomous interventions. The amount of avoided collisions increases for almost every dangerous scenarios. An important point to take into consideration while developing such systems is the increased potential of false warnings for causing a road accident due to autonomous interventions being more invasive than warnings [14].

In practice, autonomous interventions do not replace warnings and notifications given to the end-user but complement them by preparing the vehicle for taking a collision-avoidance action[4]. They also perform this action when the end-user does not react appropriately to the given warning [57].

[4] This preparation includes getting the brakes ready for an emergency brake. This kind of systems can also apply an adequate brake pressure for the current situation, even if the driver does not press the brakes hard enough.

5.5 Driving Test Results

In order to validate the simulation results and to test the portability of the approaches from the simulation into the prototype vehicles, the algorithm corresponding to the ETSI-based approach is ported to ROS and evaluated as described in the following subsections. The hardware setup on the prototype vehicles was already described in section 4.2. The corresponding software remains active on both prototype vehicles between the different trials.

Test Track

The CARISSMA Test Track is the selected test ground for validating the algorithms using prototype vehicles. Figure 5.18 shows an aerial view of the test track accompanied by its dimensions.

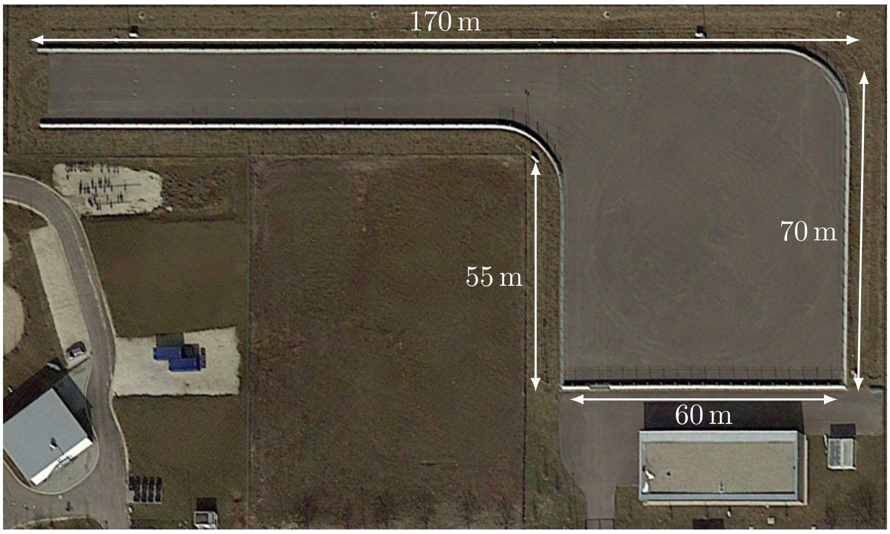


Figure 5.18: Aerial view and dimensions of CARISSMA Test Track in Ingolstadt. Image source: [58]

Two roads are built inside of the test track using traffic cones. These roads had two lanes in every direction and intersect each other on the northern section of the test track, thus building a T-junction. The eastern part of the test track is used for building non-intersecting roads. Those are used for the *Free-ride* category as well as an acceleration lane for the vehicles to approach the intersection at an adequate speed.

Scenarios

In order to cover the use cases presented in section 3.2, the following four categories of scenarios are evaluated on the test track:

- **Other-vehicle coming from right**: for validating the IMA use cases "Straight crossing path", "Left turn into crossing" and "Other-vehicle turning left from right".

- **Other-vehicle coming from left**: for validating the IMA use cases "Straight crossing path" and "Left turn into crossing".
- **Oncoming-traffic**: for validating the notifications given for LTA use cases. In addition to the two use cases explained in section 3.6, two scenarios are tested in which a vehicle waits at the intersection before turning left.
- **Free-ride**: for validating the robustness against false warnings. In such scenarios, both vehicles drive freely on the test track without any sort of coordination. This category covers two cases in which both vehicles are driving and in which one vehicle remains static.

Due to the size of the test track, and in order to ensure the safety of the drivers, the speed for this driving test is limited to 40 km/h. Every use case is evaluated using the driving speeds for the Ego-vehicle and the Other-vehicle specified in the following table:

Ego-vehicle	**Other-vehicle**
40 km/h	40 km/h
40 km/h	20 km/h
20 km/h	40 km/h
20 km/h	20 km/h

Table 5.4: Trial speeds for driving test. The only exceptions to this arrange are the category *Free-ride* and the LTA uses cases in which one vehicle remain static.

Results

Table 5.5 presents an overview of the results obtaining during the driving test. For all IMA use cases, two trials are performed with every speed configuration and for both a dangerous and a non-dangerous situation, amounting to a total of 16 trials for each one of the categories "Other-vehicle coming from left" and "Other-vehicle coming from right". For all these trials, both the prototype car and motorcycle did receive the IMA-warnings while approaching the intersection under dangerous scenarios.

The scenarios corresponding to LTA use cases in which both vehicles actively approach the intersection are evaluated two times for every speed configuration. Only two speed configurations are evaluated for the LTA scenarios in which one vehicle waits before turning left, i.e. two different speeds for the vehicle approaching the intersection. For all this trials, both the prototype car and motorcycle are able to receive the LTA-notifications while intending to turn left at the intersection.

The *Free-ride* trial in which only one vehicle is moving yields a true negatives rate of 100 %, i.e. no false warning is received. For the cases in which both vehicles are moving and the risk of collision is relatively high, a warning is given to both of them and directly followed by a braking maneuver. The vehicles also receive some warnings in non-dangerous situations. In all such situations, the vehicles are driving in opposite directions and following a curve.

Use Case	Scenario	Trials	TP	TN
IMA	Straight crossing path	16	✓	✓
IMA	Left turn into crossing	16	✓	✓
IMA	Other-vehicle turning left from right	16	✓	✓
LTA	Left turn with oncoming-traffic	8	✓	✓
LTA	Other-vehicle turning left ahead	8	✓	✓
LTA	Waiting before turning left	4	✓	✓
LTA	Waiting before turning left (after half turn)	4	✓	✓
IMA/LTA	Free-ride (Both vehicles moving)	1	✓	✗
IMA/LTA	Free-ride (Only one vehicle moving)	1	–	✓

Table 5.5: Overview of driving test results. The rightmost columns stand for true positives and true negatives. The former refers to the algorithm recognizing a dangerous situation and issuing a warning over the HMI while the latter indicates if false warnings are given.

5.6 Discussion

This chapter describes the scenarios and criteria for the evaluation of the algorithms. All four algorithms are implemented and evaluated inside of the simulation. The ETSI-based approach is also implemented on the prototype vehicles for validation of the simulation results.

For the evaluation inside of the simulation environment, a data set covering relevant accident scenarios is generated, while the different driver reactions to warnings are ordered into six different models, every one having a certain probability of occurrence. Also the different indicators of performance are defined based on the notion of true/false positives and negatives. This concept is expanded in order to differentiate between collisions being detected and avoided, and collisions being detected but not avoided, e.g. due to a too-late detection.

The results obtained through the simulation are then presented. This includes the detection accuracy, the robustness against false warnings and the runtime for the four algorithms. In general, three scenarios are specially challenging for all four algorithms: 351, 215 and 306. The first two correspond to LTA use cases in which the Other-vehicle turns left at an intersection, and the third one pictures the Other-vehicle turning right while approaching the Ego-vehicle from the left. What makes these scenarios specially challenging for the algorithms? The answer lies in the maneuver to be performed by the Other-vehicle being unknown to the algorithm.

The results obtained through the simulation are then validated using the prototype vehicles. For these purposes, the different use cases for IMA and LTA are emulated on a test track. The implemented algorithm is able to correctly distinguish between dangerous and non-dangerous situations for all reproduced use cases. It nevertheless failed to constantly deliver true negatives for the category *Free-ride* while both vehicles are moving. This is specially appreciated as both vehicles drive in opposite directions while approaching a curve, thus confirming the results gathered through the simulation.

All four algorithms are also observed to have a linear time complexity $O(n)$ for processing n number of CAMs. And a space complexity corresponding to $O(1)$, because of no information about past messages being stored. The potential of autonomous interventions is evaluated by adding a 7^{th} reaction model. This model represents a collision-avoidance action being performed without having to take the driver reaction time into account.

6 Summary and Final Words

6.1 Summary

The great risk of being involved in a road accident becomes clear when looking at current accident figures, especially at intersections. Different measures have been taken by the automotive industry in order to avoid those accidents or mitigate their consequences. Most of the solutions for the prevention of collisions are built exclusively upon on-board sensors. This technology, however, has proven to be insufficient for greatly preventing accidents at intersections when there is no Line-of-Sight between the vehicles, which leads to the too-late detection of crossing-traffic. This becomes especially critical for the cases including motorcycles, as their figure is relative slim in comparison to cars and their detection by conventional on-board sensors becomes more difficult. A relatively new vehicular technology having a great potential for changing this picture is the V2V communication. This promising technology aims to create a network for the vehicles to exchange information wirelessly, thus being able to be aware of each other even when conventional on-board sensors cannot yet perceive them.

This work targets the challenges associated to the development of an ADAS based on this technology, with a special focus on intersections as its working area. It takes the currently available technologies and standards as the starting point for the conception and implementation of this safety-critical application. It also has the goal of producing a system that works independently of the road infrastructure and does not rely on other vehicles implementing complex, optional aspects of the standards.

Based on these considerations, this work is structured in three main chapters: the conception and requirements for the application, the technical implementation in both a simulation environment and prototype vehicles, and an assessment of its potential for the prevention of collisions.

The chapter *Conception* states the boundaries and requirements when using modern technologies for communication, localization and context awareness. It describes four different approaches for the prediction of collisions. The first one aims to prevent the vehicles from entering an intersection when there is a high probability of other vehicle also entering this same intersection (SAE). The second approach, the only one not using map data, intends to estimate a Point-of-Collision based on the travel direction of the vehicles and the probability of them reaching this point at approximately the same time (ETSI). The third approach, being an expansion of the second one, uses map data and the position of the vehicles for estimating the Point-of-Collision (ETSI-Map). The fourth approach is the only approach using a discrete method for the risk computation, it predicts the future position of the vehicles and searches for any overlap, which is an indicator for a high risk of collision (Ghost vehicles).

This chapter also presents the real-time characteristics required by the application in order to be useful. These characteristics raise the question about which operation should be performed first: the verification of digital signatures or the computation of the risk of a collision. This is because the verification of signatures can take longer than the risk computation, which could lead to some received CAM not being processed. The order exchange intends to guarantee that all incoming information is processed accordingly. Another important point mentioned is the warning dilemma, which states that the acceptance of such an application is directly coupled to its rate of false warnings, thus making a high accuracy in the prediction of collisions mandatory.

The chapter *Implementation* then introduces two different kind of fields for the implementation of this application: a simulation environment and prototype vehicles. The simulation environment offers a safe method for emulating dangerous road situations in several different configurations. This is exploited for optimizing the lowest threshold of the parameters TTC and Encroachment time. The initial idea is the use of a fixed lowest threshold for the TTC. This has proven inadequate for addressing roads having different speed limits because of a low threshold value issuing too-late warnings in the upper speed ranges and a high threshold value providing too-early warnings for the lower speed ranges. Another task completed using the simulation is the search for optimal values of the lowest threshold of the Encroachment time. It is observed that this parameter directly influences the amount of true positives and true negatives. A high threshold value leading to an increase in the rate of true positives, i.e. more collisions are detected, but at the same time reducing the rate of true negatives, i.e. more false warning are likely to be given. A low threshold value produces the contrary effect.

A total of five algorithms are presented in this chapter. Four corresponding to the implementation of the presented approaches and one for the extraction of the map data. This algorithm uses the position and orientation of the vehicles for determining if they are approaching the same intersection, and if they do, returning information related to it. An alternative to this procedure, which is considered to have a much lower execution time, is using the position and heading of the Ego-vehicle in order to get information about the next intersection ahead. This information would then be stored and used upon the reception of every CAM until the Ego-vehicle passes this intersection or finishes the trip.

The chapter *Evaluation* then presents an evaluation of the different approaches and compares their potential for correctly detecting dangerous and non-dangerous situations. These results revealed three scenarios which pose an special challenge for all approaches. One of these scenarios corresponds to an oncoming-traffic situation in which the Other-vehicle turns left at the intersection while the Ego-vehicle intends to drive straight across it. The challenge lies in the Ego-vehicle not being aware of the Other-vehicle's maneuver, as it is not mandatory for the Other-vehicle to broadcast such an information. A way of preventing accidents under this scenario consists in the Other-vehicle also being equipped with an equivalent collision-avoidance system. Another challenging scenario arises when both vehicles safely turn left in front of each other, which tends to produce false warnings. Such scenarios are observed to mostly take place on broad, signalized intersections. This meaning that those scenarios can be identified, and subsequently filtered out, by using map data. The third challenging scenario is the Other-vehicle turning left while approaching the Ego-vehicle from the left. Here, the lack of information about the Other-vehicle's maneuver is a source of false warnings. Lowering the threshold for the Encroachment time could be used for decreasing the rate of false positives in such cases. Conversely, right-before-left warnings can be reliably implemented using the presented methods.

Characteristics	Approaches			
	SAE	ETSI	ETSI_{Map}	Ghost
Accuracy of true warnings for IMA	+	++	++	+
Accuracy of notifications for IMA	++	+	+	+
Accuracy of true warnings for LTA	+	−−	++	+
Accuracy of notifications for LTA	++	−−	+	+
Robustness against false warnings	+	−−	+	++
Independence of intersection geometry	−	++	+	+
Scalability regarding size of intersection	+	++	+	−

Table 6.1: Comparison of the four different approaches.

Table 6.1 presents a comparison of the main characteristics across all four approaches. ETSI-Map stands out as the more suitable approach for the implementation of IMA and LTA. The obtained results also emphasize that the use of map data is necessary for avoiding false warnings and mandatory for covering LTA use cases (see contrast between ETSI and ETSI-Map). They also reveal that the overall robustness against false warnings would not be high enough for allowing such an application to issue warnings directly. The information provided by the application needs to be contrasted with the readings of other sensors before issuing any warnings to the end-user. This applications should then be seen as an additional sensor delivering information to a central controller device, which also gathers information from other on-board devices and determines the need and the form of the warnings to be given.

6.2 Final Words and Future Work

The presence of connected ADAS is essential in a future world, in which the roads become more complex, most road users are able to communicate with each other and several vehicles even drive autonomously along the streets. This work identifies and addresses several technical challenges and intends to contribute to the advancement of such systems. Motorcycles, as part of the group of VRUs, can profit enormously from their implementation, as they have the potential of increasing the awareness of other road users and also autonomous vehicles.

From the very start, the framework developed for this work is conceived for allowing others to build upon its base. The used simulation environment, for example, supports the development of systems combining conventional sensors and V2X-communication, this one being one of the main reasons for its selection.

Which are the ideal complementary sensors and which method best fuse their readings with a V2X-application are important questions still to be answered. Also the challenge of reducing the necessary hardware into a single control device and the usage of machine learning techniques for collision prediction constitute some of the next steps forward.

7 Appendix

7.1 Fields Contained in Cooperative Awareness Messages (CAMs)

Container	Attribute		Definition
Basic Container	stationType	M	Type of road user
	referencePosition	M	Longitude, latitude and altitude
High-Frequency Container	heading	M	Heading with regards to the true north
	speed	M	Speed value in cm/s
	driveDirection	M	Forward or backwards
	vehicleLength	M	Vehicle length in dm
	vehicleWidth	M	Vehicle width in dm
	longitudinalAcceleration	M	Acceleration in dm/s^2
	curvature	M	Turning radius in $m/30000$
	curvatureCalculationMode	M	Indicates if yaw rate is used for calculation
	yawRate	M	Vehicle rotation around z-axis in $0.01°/s$
Low-Frequency Container	vehicleRole	O	Role played by a vehicle
	exteriorLights	O	Status of exterior light switches
	pathHistory	O	Set of path points
Special Vehicle Container	publicTransportContainer	C	Defined in ETSI EN 302 637-2 [24]
	specialTransportContainer	C	Defined in ETSI EN 302 637-2 [24]
	dangerousGoodsContainer	C	Defined in ETSI EN 302 637-2 [24]
	roadWorksContainerBasic	C	Defined in ETSI EN 302 637-2 [24]
	rescueContainer	C	Defined in ETSI EN 302 637-2 [24]
	emergencyContainer	C	Defined in ETSI EN 302 637-2 [24]
	safetyCarContainer	C	Defined in ETSI EN 302 637-2 [24]

Table 7.1: Contents of CAM containers. *M* stands for *mandatory*, *O* stands for *optional* and *C* stands for *conditional*. Attributes are specified in [25, 24].

7.2 Sub-Channelization for LTE-V

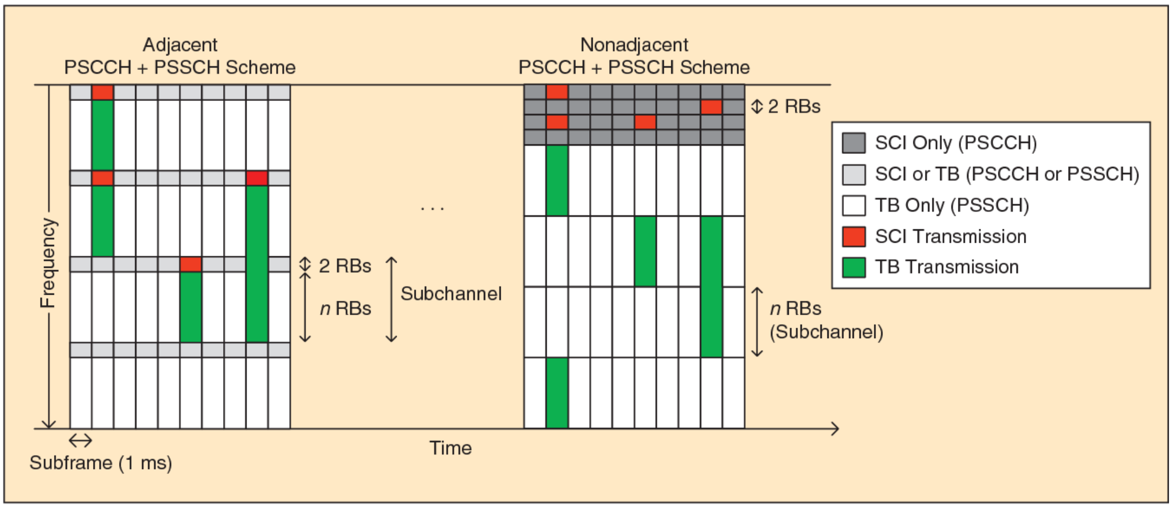


FIGURE 1 LTE-V subchannelization.

Figure 7.1: Sub-channelization strategies for LTE-V. Source: [59].

Figure 7.1 shows two strategies for sub-channelization used by LTE-V. Those are also used by Cellular-V2X and consist of selecting one or multiple sub-channels of a sub-frame in order to send a single V2X-message.

Based on current standards, the channels can have a width of either 10 MHz or 20 MHz. The sub-channels are organized in Resource Blocks (RBs) of 180 kHz, which are composed by 12 sub-carriers of 15 kHz each. Every sub-carrier is able to transport 14 symbols. From those, only 9 are dedicated to the transmission of the message while the others are used for modulation and guard. Two RBs are needed for sending the Sidelink Control Information (SCI), which contains information about the succeeding Transport Block (TB).

7.3 Clustering of Accident Types

The following are the accident codes corresponding to the categories of accidents which are targeted by IMA and LTA:

- **Type 2**: 211, 212, 213, 214, 215, 261, 262, 271, 281.
- **Type 3**: 301, 302, 303, 311, 312, 313, 321, 322, 323, 331, 332, 333, 351, 352, 353, 354, 355.

Those accident codes were clustered in different categories according to the maneuvers of the vehicles and are independent of the right-of-way:

Application (Situation)	Maneuver	Accident codes
IMA (OV comes from left)	EV driving straight	301, 311
	EV turning lef	302, 312, 261
	EV turning right	303, 313
IMA (OV comes from right)	EV driving straight	321, 331, 355, 353, 271
	EV turning left	322, 332, 352
	EV turning right	323, 333
LTA (EV turns left)	OV driving straight	211, 281, 354
	OV turning right	212
	OV turning left	215

Table 7.2: Clustering of accident codes in different categories of scenarios.

7.4 Operation of *scenario_ builder.py*

The *scenario_ builder.py* offers the following options for tuning the simulation environment in order to develop and test different algorithms:

- **Selection of use cases**: those can be chosen based on their name (e.g. 302-30c) or type (e.g. 302). A small set of use cases is found in the file "scenario_config_file.xml".
- **Selection of algorithm**: all four implemented approaches are available and are added to the Other-vehicle specified in the configuration file. In order to test scenarios without any algorithm intervention, the value "none" can be passed to the corresponding argument.
- **Selection of driver reaction model**: A total of eight models can be selected: 1-6 based on real driver reactions, an ideal model and a no-reaction model.
- **Fixed value for** TTC_{min} / TTI_{min} / t_{pred}: this fixes the minimum threshold of the TTC-related parameter, i.e. the threshold does not depend on the speed anymore.
- **Fixed value for** t_{enc_min} / d_{min}: this fixes the minimum threshold of Encroachment time or distance.
- **Disable visual aid for debugging**: by setting this option, the lines and boundary boxes generated by the simulation environment are disabled. The visual indications for "Notification", "Warning" and "Braking" remain active.
- **Disable addition of GNSS noise**: this option simulates an ideal GNSS receiver.
- **Scan values of parameters**: this is thought as a tool for automatically testing several configurations of fixed TTC_{min} and t_{enc_min} values. A range for every parameter must be provided, which indicates the initial value, the final value and the value steps to be used.
- **Generate report**: this option generates a report containing the scenario(s), the speed of the vehicles, the expected outcome, the driving model used and the result.

7.5 Computer Hardware for Simulation Environment

Type	Notebook
Processor	Intel Core i7-6820HQ (8x 2.7 GHz)
RAM-Memory	32 GB
Graphic Card	Quadro M2000
Operating System	Ubuntu 18.04
Kernel	5.4.0-42

Table 7.3: Characteristics of computer hardware for simulation environment.

Bibliography

[1] Statistisches Bundesamt (Destatis), "Kraftrad- und Fahrradunfälle im Straßenverkehr 2018." https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/unfaelle-zweirad-5462408187004.pdf?__blob=publicationFile, 2019.

[2] Statistisches Bundesamt (Destatis), "Verkehrsunfälle - Fachserie 8 - 2018." https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-jahr-2080700187004.pdf?__blob=publicationFile, 2019.

[3] R. Volpe, ed., *Casebook of Traumatic Injury Prevention*. Springer International Publishing, 2020.

[4] Mobility and Transport - European Commission, "ITS & Vulnerable Road Users." https://ec.europa.eu/transport/themes/its/road/action_plan/its_and_vulnerable_road_users_en. (accessed: 22.06.2020).

[5] E. R. Teoh, "Effectiveness of antilock braking systems in reducing motorcycle fatal crash rates," *Traffic injury prevention*, vol. 12,2, 2011.

[6] Volkswagen AG, "Car2X im neuen Golf: Ein „technischer Meilenstein" (in German)." https://www.volkswagen-newsroom.com/de/storys/car2x-im-neuen-golf-ein-technischer-meilenstein-5919, 2020. (accessed: 21.06.2020).

[7] Cadillac Pressroom, "V2V Safety Technology Now Standard on Cadillac CTS Sedans." https://media.cadillac.com/media/us/en/cadillac/news.detail.html/content/Pages/news/us/en/2017/mar/0309-v2v.html, 2017. (accessed: 20.07.2020).

[8] Andreas von Eichhorn, *Querverkehrassistentz unter Berücksichtigung von Unsicherheiten aus Sensorik und Prädiktion*. PhD thesis, Universität Duisburg-Essen, 2014.

[9] 5G Carmen, "Use Cases and Requirements." https://5gcarmen.eu/wp-content/uploads/2020/03/5G_CARMEN_D2.1_FINAL.pdf, 11 2018.

[10] Vittorio Astaritaa, Demetrio Carmine Festaa, Vincenzo Pasquale Giofrèa, Giuseppe Guido, "Surrogate Safety Measures from Traffic Simulation Models a Comparison of different Models for Intersection Safety Evaluation," 9 2018.

[11] K. Nobukawa, *A Model Based Approach to the Analysis of Intersection Conflicts and Collision Avoidance Systems*. PhD thesis, University of Michigan, 2011.

[12] A. V. Maarten Löffler, "Lecture notes in collision detection." http://www.cs.uu.nl/docs/vakken/ddm/2015-2016/Lecture%209%20-%20Collision%20Detection.pdf, 2016.

[13] M. Rahman, *Driver acceptance of advanced driver assistance systems and semi-autonomous driving systems.* PhD thesis, Mississippi State University, 2016.

[14] Florian Schellin, "Sensitivitätsanalyse eines AS-Systems unter Berücksichtigung von V2X-Kommunikation zur Vermeidung von Ausgewählten PKW-KRAD-Unfällen," Master's thesis, Technische Universität Berlin, 3 2018.

[15] Kraftfahrt-Bundesamt, "Personenkraftwagen am 1. Januar 2020 nach ausgewählten Merkmalen." https://www.kba.de/DE/Statistik/Fahrzeuge/Bestand/Jahresbilanz/fz_b_jahresbilanz_thema_im_Ueberblick/2020_b_barometer.html?nn=2598042, 2020. Tab: *Segmente* (accessed: 02.07.2020).

[16] Stéphanie Lefèvre, Jonathan Petit, Ruzena Bajcsy, Christian Laugier, Frank Kargl, "Impact of V2X privacy strategies on intersection collision avoidance systems," in *IEEE Vehicular Networking Conference*, (Boston, United States), 2013.

[17] ETSI, "Pre-standardization study on pseudonym change management - ETSI TR 103 415," tech. rep., ETSI, 4 2018.

[18] Mobility and Transport - European Commission, "Cooperative, connected and automated mobility (CCAM)." https://ec.europa.eu/transport/themes/its/c-its_en. (accessed: 24.06.2020).

[19] 3rd Generation Partnership Project (3GPP), "Initial Cellular V2X standard completed," tech. rep., 3rd Generation Partnership Project (3GPP), 2016.

[20] Institute of Electrical and Electronics Engineers (IEEE), "802.11p-2010 - IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," tech. rep., Institute of Electrical and Electronics Engineers (IEEE), 2010.

[21] ETSI, "Specifications of Decentralized Environmental Notification Basic Service - ETSI EN 302 637-3," tech. rep., ETSI, 9 2014.

[22] ETSI, "Intersection Collision Risk Warning (ICRW), application requirements specification - ETSI TS 101 539-2," tech. rep., ETSI, 6 2018.

[23] SNIP, "SAE J2735 DSRC Message List." https://www.use-snip.com/kb/knowledge-base/sae-j2735-dsrc-message-list/. (accessed: 25.06.2020).

[24] ETSI, "Specification of Cooperative Awareness Basic Service - ETSI EN 302 637-2," tech. rep., ETSI, 9 2014.

[25] ETSI, "Applications and facilities layer, common data dictionary - ETSI TS 102 894-2," tech. rep., ETSI, 8 2018.

[26] CAR 2 CAR Communication Consortium, "Survey on ITS-5G CAM statistics," tech. rep., CAR 2 CAR Communication Consortium, 12 2018.

[27] M. Hasan, S. Mohan, T. Shimizu, and H. Lu, "Securing Vehicle-to-Everything (V2X) Communication Platforms," 03 2020.

[28] ETSI, "ITS communications security architecture and security management - ETSI TS 102 940," tech. rep., ETSI, 4 2018.

[29] Autotalks, *CRATON2/SECTON Security - Application Note*, 2019. Rev. 1.1.

[30] Mobility and Transport - European Commission, "Deployment and Operation of European Co-operative Intelligent Transport Systems (C-ITS)." https://ec.europa.eu/transport/sites/transport/files/c-its_certificate_policy_release_1.pdf.

[31] C. Rizos, "Multi-Constellation GNSS/RNSS from the Perspective of High Accuracy Users in Australia," *Journal of Spatial Science*, vol. 53, 12 2008.

[32] United States Department of Transportation, "Enhanced Digital Mapping Project: Final Report," tech. rep., United States Department of Transportation, 11 2004.

[33] u-blox, *u-blox M8 untethered dead reckoning module including 3D inertial sensors*, 6 2020. Rev. 08.

[34] F. Jimenez, ed., *Intelligent Vehicles - Enabling Technologies and Future Developments*. Butterworth-Heinemann, 2017.

[35] S. Liu, ed., *Engineering Autonomous Vehicles and Robots: The DragonFly Modular-based Approach*. Wiley-IEEE Press, 2020.

[36] D. L. Fisher, W. J. Horrey, J. D. Lee, and M. Regan, eds., *Handbook of Human Factors for Automated, Connected, and Intelligent Vehicles*. CRC Press, 2020.

[37] Autobild, "Was taugen Totwinkelwarner?." https://www.autobild.de/bilder/totwinkelwarner-im-test-3920286.html#bild1, 3 2013. (accessed: 24.07.2020).

[38] ADAC, "ADAC Crashtest - kleiner Crash mit großen Folgen." https://www.adac.de/infotestrat/tests/crash-test/bagatell_crash, 9 2013. (accessed: 24.07.2020).

[39] Arthur Werle, "Optimierung von Fahrerwarnungen mittels Probandenversuchen am Motorrad-Simulator," Master's thesis, Technische Universität Berlin, 3 2020.

[40] Mobility and Transport - European Commission, "Current speed limit policies." https://ec.europa.eu/transport/road_safety/specialist/knowledge/speed/speed_limits/current_speed_limit_policies_en. based on studies made by consultants (accessed: 08.07.2020).

[41] I. T. S. Committee, "IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages ," tech. rep., IEEE Vehicular Technology Society, 2016.

[42] SAE, "On-Board System Requirements for V2V Safety Communications (J2945/1_201603)," tech. rep., SAE, 3 2016.

[43] THINK!, ""Take longer to look for motorbikes" poster." https://www.think.gov.uk/campaign/motorcycling/. (accessed: 24.07.2020).

[44] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[45] Vernetzte Welt, “BMW R 1200 RS Connected Ride: Der vernetzte Trendsetter.” http://vernetzte-welt.com/bmw-r-1200-rs-connected-ride-der-vernetzte-trendsetter/, 11 2017.

[46] Auto Zeitung, “BMW X5 xDrive40e 2015: SUV mit Plug-in-Hybrid-Antrieb.” https://www.autozeitung.de/bmw-x5-xdrive40e-2015-plug-in-hybrid-93503.html?image=8, 2015.

[47] Charles F. F. Karney, “Algorithms for geodesics.” https://link.springer.com/content/pdf/10.1007/s00190-012-0578-z.pdf, 2013.

[48] VUFO GmbH, “Warning time for C-ITS systems under longitudinal- and crossing-traffic conditions,” 9 2019. Preliminary.

[49] S. Lefevre, C. Laugier, and J. Ibanez-Guzman, “Exploiting map information for driver intention estimation at road intersections,” 06 2011.

[50] Felix Klanner, *Entwicklung eines kommunikationsbasierten Querverkehrsassistenten im Fahrzeug.* PhD thesis, Technische Universität Darmstadt, 2008.

[51] Gesamtverband der Deutschen Versicherungswirtschaft e. V. , “Unfalltypen-Katalog.” https://udv.de/sites/default/files/tx_udvpublications/unfalltypen-katalog_udv_web_2.pdf, 1 2016.

[52] Hagen Schüller, “Modelle zur Beschreibung des Geschwindigkeitsverhaltens auf Stadtstraßen und dessen Auswirkungen auf die Verkehrssicherheit auf Grundlage der Straßengestaltung .” https://tu-dresden.de/bu/verkehr/ivs/vnm/ressourcen/dateien/institutsschriftenreihe/Heft-12.pdf, 12 2010.

[53] N. Kudarauskas, “Analysis of emergency braking of a vehicle,” *Transport*, vol. 22, 07 2007.

[54] simTD, “Simulation realer Verkehrsunfälle zur Bestimmung des Nutzens für ausgewählte simTD-Anwendungsfälle auf Basis der GIDAS Wirkfeldanalyse,” tech. rep., European Center for Information and Communication Technologies (EICT), 1 2013.

[55] Leonardo Lo Schiavo, “Performance of MEC Solutions in Automotive Applications,” Master’s thesis, Politecnico Di Torino, 10 2018.

[56] Mark Alexander Mages, *Top-Down-Funktionsentwicklung einer Einbiege- und Kreuzenassistenten.* PhD thesis, Technische Universität Darmstadt, 2008.

[57] European New Car Assessment Programme (Euro NCAP), “Notbremssystem Fahrzeug-Fahrzeug.” https://www.euroncap.com/de/fahrzeugsicherheit/die-bedeutung-der-bewertungen/assistenzsysteme/notbremssystem-fahrzeug-fahrzeug/, 2020.

[58] G. Maps, “CARISSMA Outdoor-Freiversuchsgelände.” https://goo.gl/maps/rt8qwpVZCWxmM2Wh7. (accessed: 02.08.2020).

[59] R. Molina-Masegosa and J. Gozálvez, “LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications,” *IEEE Vehicular Technology Magazine*, vol. 12, pp. 30–39, 2017.