Modelling and Simulation for Evaluation of Cooperative Intelligent Transport System Functions

Maytheewat Aramrattana
Abstract

Future vehicles are expected to be equipped with wireless communication technology, that enables them to be “connected” to each others and road infrastructures. Complementing current autonomous vehicles and automated driving systems, the wireless communication allows the vehicles to interact, cooperate, and be aware of its surroundings beyond their own sensors’ range. Such systems are often referred to as Cooperative Intelligent Transport Systems (C-ITS), which aims to provide extra safety, efficiency, and sustainability to transportation systems. Several C-ITS applications are under development and will require thorough testing and evaluation before their deployment in the real-world. C-ITS depend on several sub-systems, which increase their complexity, and makes them difficult to evaluate.

Simulations are often used to evaluate many different automotive applications, including C-ITS. Although they have been used extensively, simulation tools dedicated to determine all aspects of C-ITS are rare, especially human factors aspects, which are often ignored. The majority of the simulation tools for C-ITS rely heavily on different combinations of network and traffic simulators. The human factors issues have been covered in only a few C-ITS simulation tools, that involve a driving simulator. Therefore, in this thesis, a C-ITS simulation framework that combines driving, network, and traffic simulators is presented. The simulation framework is able to evaluate C-ITS applications from three perspectives; a) human driver; b) wireless communication; and c) traffic systems.

Cooperative Adaptive Cruise Control (CACC) and its applications are chosen as the first set of C-ITS functions to be evaluated. Example scenarios from CACC and platoon merging applications are presented, and used as test cases for the simulation framework, as well as to elaborate potential usages of it. Moreover, approaches, results, and challenges from composing the simulation framework are presented and discussed. The results shows the usefulness of the proposed simulation framework.
Acknowledgments

For their encouragement and patience, I would like to express my gratitude to my main supervisor, Prof. Tony Larsson at Halmstad University, and co-supervisor, Dr. Jonas Jansson at the Swedish National Road and Transport Research Institute (VTI). I also would like to thank my mentor at VTI, Arne Nåbo. They have given me this opportunity and endless support. Apart from Halmstad University and VTI, this work has been supported by SAFER¹ and the Knowledge Foundation (KKS) through the Vehicle ICT Innovation Methodology (VICTIg) project.

I am grateful and lucky to be part of many working environment such as the Embedded and Intelligent Systems Industrial Graduate School (EISIGS), the School of Information and Technology at Halmstad University, and the SIM (Driving simulation and visualization) unit at VTI. I would like to thank all my colleagues there for fruitful discussions and their assistance. Special thanks to the Halmstad GCDC 2016 team and everyone involved for giving me priceless experiences.

Last but not least, I would like to thank my family for their unconditional love and support. Also, all my friends for being there and backing me up in life and work.

¹Vehicle and Traffic Safety Centre at Chalmers (SAFER)
List of Publications

This thesis summarizes the following publications.


# Contents

1 Introduction ........................................ 1  
   1.1 Motivation, Purposes, and Goals .............. 2  
   1.2 Research Questions .......................... 4  

2 Related Works and Background ................. 5  
   2.1 Related Works ................................ 7  
      2.1.1 Modelling and Simulation of C-ITS .... 7  
      2.1.2 State of the art ......................... 8  
   2.2 Background ................................... 11  

3 Contributions .................................... 15  
   3.1 Summary of Paper I ........................... 16  
   3.2 Summary of Paper II .......................... 18  
   3.3 Summary of Paper III ......................... 21  

4 Conclusions ...................................... 23  
   4.1 Discussion .................................... 24  
   4.2 Future Works .................................. 27  

References ........................................ 29  

A Paper I ........................................... 37  

B Paper II .......................................... 45  

C Paper III ......................................... 51
List of Figures

1.1 Essential models in C-ITS simulation ......................... 3

2.1 Details of levels of automation presented by SAE [4] ........ 6
2.2 The first order low-pass filter applied to the vehicles in plexe-sumo. 12
2.3 The “Sim IV”, driving simulator with motion system at VTI .... 12

3.1 The composition structure of the simulation framework for modelling of C-ITS. ................................. 16
3.2 Dimensions of cooperative ITS .................................. 17
3.3 Overview of the extended driving simulator ..................... 19
3.4 Platoon of vehicles with parameters .............................. 20
3.5 Overview of the simulated platoon merging scenario ........... 22

4.1 Comparison of the speed profile when accelerate from 0 to 30 km/h. ...................................................... 25
List of Tables

2.1  Levels of driving automation proposed by SAE and NHTSA . . 5
2.2  Network parameters in *plexe-veins*  . . . . . . . . . . . . . . . . 11
Chapter 1
Introduction

With aims to provide safe and comfortable driving, various advanced driver assistance systems (ADAS) has been developed over the previous decades. They offer a wide range of services such as anti-lock braking systems (ABS), lane keeping assist (LKA), blind spot information system (BLIS), cruise control (CC), adaptive cruise control (ACC), etc. Modern vehicles are equipped with sensors such as radar, ultrasonic, camera, light detection and ranging (LIDAR), etc., and Global Navigation Satellite System (GNSS) receivers such as a Global Positioning System (GPS) receiver, to support the operation of ADAS. Connecting these systems and capabilities to control the actuators of the vehicles, evolves towards automated driving systems, where vehicles are able to navigate themselves without a human driver involved. Many autonomous vehicles have been developed [5, 6, 7, 8], and some have also been driving in real traffic [9, 10, 11]. Furthermore, developments in wireless communication enable vehicles to be connected, both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). The communications provide information about the surroundings beyond the range of sensors. For instance, being aware that the vehicle in front of the preceding vehicle is braking at its maximum power, allows the ego vehicle to start braking early to avoid or mitigate a severe rear-end collision.

Consequently, connected and automated driving concepts have been introduced in the context of cooperative intelligent transport system (C-ITS). C-ITS incorporates information and communication technologies into the transport systems. C-ITS strives for safer, more efficient, and more sustainable transport systems. Being connected will increase the awareness of vehicles about their surroundings. To achieve the goals and improve transport systems, interaction and cooperation between actors are key factors. To enable this, reliable communication is required, since vehicles driving in automated mode can only exchange information with each other through wireless communication. Ultimately, those key factors and reliability increase the complexity of the system, which needs to be tested and evaluated.
CHAPTER 1. INTRODUCTION

Compared to ADAS, C-ITS is a relatively new technology. Standards and guidelines related to C-ITS have not yet been well-established. Vehicles in C-ITS can be seen as systems, interacting with each others and a larger common system forming a C-ITS. Therefore, C-ITS can also be considered as a system of systems. To reach its maturity, further development and extensive evaluation of the system are required. However, to properly evaluate the system, at least two actors are necessary. In early development phases, products that support C-ITS are costly to build, because it has not yet been produced regularly. Moreover, some legally regulated equipments or systems may not yet exist. Therefore, simulation is a suitable technique to support development and testing of C-ITS. It has proper characteristics to support design and evaluation of systems such as: \(a\) safety, e.g. dangerous or high-speed scenarios can be performed in simulation without risk to cause any harm; \(b\) cost-efficient, e.g. real vehicles or test-beds are often not needed; \(c\) flexibility, e.g. it is easier to change the structure of simulated systems; and \(d\) repeatability, e.g. executing exactly the same scenarios are often possible, and enhances statistical analyses.

These characteristics listed above are important for testing and evaluation of C-ITS especially in its early development stages. For example, simulation can help developers of functions in C-ITS to test their ideas without a need for real vehicles. Therefore, with the aim to support development and evaluation of C-ITS, this thesis presents a tool for modelling and simulation of C-ITS.

1.1 Motivation, Purposes, and Goals

C-ITS is a new paradigm in transportation systems with a lot of potentials. A safer, more efficient, and more sustainable transportation system can be achieved with C-ITS through interaction and cooperation between actors in the system. Consequently, there are dependencies between the different actors in the system. With increasing dependencies, complexity of the system also grows, and makes it more difficult to test and evaluate the system. Going through all possible scenarios in C-ITS is very difficult, it is almost impossible. Studying C-ITS requires interdisciplinary knowledge. Existing methodologies and tools from only one research area are no longer feasible to efficiently test and evaluate C-ITS. A novel tool and methodology, or a combination of existing ones are needed. Especially simulation tools, since a complete C-ITS platform might not be available yet. Besides, several subsystems in C-ITS are also in their early development phase, and the specification of the subsystems is not yet settled. It would be costly to build a realistic test-bed following such uncertain requirements. Therefore, as a part of the Vehicle ICT Innovation Methodology (VICTIg) project funded by Knowledge Foundation (KKS) and SAFER, a goal is to develop methodology for evaluation of C-ITS functions, including the functional safety aspects of them. The methodology aims to support researchers and developers in design and evaluation of C-ITS functions during early development phases.
Major components of C-ITS that need to be modelled are illustrated in Fig. 1.1. Typically, existing simulation tools are specialized in modelling some parts or aspects of the system. For instance, a driving simulator is excellent at providing interaction with human driver and vehicle dynamics. But, it usually do not consider V2V and V2I communication, and the surrounding traffic in driving simulators is often simplified. On the other hand, a microscopic traffic simulator can model more complex traffic in a bigger road network, if compared to driving simulators. A mixture of different traffic behaviours can be modelled in a traffic simulator. Modelling of vehicle-to-vehicle and vehicle-to-infrastructure (V2X) communication is normally handled by a network simulator. However, the nodes in the network are usually static in network simulators. To adapt to motion of the nodes in C-ITS such as vehicles, a network simulator is often coupled with a traffic simulator to simulate C-ITS scenarios, as presented in [12, 13].

Cooperative adaptive cruise control (CACC) and its applications such as platooning, are one of the first C-ITS applications expected to be deployed soon. Also, the CACC applications create many challenges and provides several benefits [14, 15, 16, 17].

For the above reasons, this thesis presents a C-ITS simulation framework that combines driving, network, and traffic simulators. CACC and its applications such as platoon merging are chosen as the first set of C-ITS functions to be evaluated using the simulation framework, they also demonstrate the capa-
Chapter 1. Introduction

The capabilities of the simulation framework. Many aspects of CACC applications can be evaluated using the framework as will be presented and discussed in this thesis.

1.2 Research Questions

The main research questions of the VICTIg project is how to develop and evaluate cooperative functions in an efficient way? Given the diverse range of the functions, this thesis will mainly focus on CACC and its applications. As aforementioned, simulation is an essential technique to support evaluation and development of C-ITS functions; so a simulation tool dedicated to C-ITS is needed. Therefore, the main research question addressed in this thesis is:

RQ 1 How to create a simulation environment for CACC evaluation?

The simulation environment is intended to address more specific research question, which is how to perform testing and get sufficient test coverage by simulation of cooperative driving situations? (e.g., involving several vehicles or road side units). Using a combination of existing simulators is chosen as an approach to answer RQ 1. Therefore, the following more specific research questions are tackled:

RQ 1.1 What need to be modelled and simulated for testing and evaluation of CACC?

RQ 1.2 What are the level of abstraction and accuracy needed in each model?

RQ 1.3 How to integrate and synchronize the simulators?

RQ 1.4 What are the interfaces required to incorporate existing models?
Chapter 2
Related Works and Background

Cooperative Automated Driving

By enabling vehicles to perceive information beyond their sensors’ range, cooperative driving can be seen as a complement and an enhancement to automated driving. In C-ITS applications, a certain degree of vehicle automation is expected, even though driving automation is not a requirement for successful cooperation. For instance, drivers interact with each others via eye contacts and body languages in today’s traffic. Driving automation provides a basis and complement in the evolution towards successful C-ITS, in order to enable safer, more efficient, and more sustainable transportation systems.

Several organizations have proposed classifications or levels of driving automation. For instance, the Society of Automotive Engineers (SAE) and National Highway Traffic Safety Administration (NHTSA) have presented their view on the levels of driving automation. Moreover, in Germany, the Federal Highway Research Institute (BASt) and German Association of the Automotive Industry (VDA) have presented similar views but with slightly different definitions. The definitions from SAE and NHTSA, presented in Table. 2.1, are the most commonly used. Please refer to the table I in the Paper I for an exhaustive list.

Table 2.1: Levels of driving automation proposed by SAE and NHTSA.

<table>
<thead>
<tr>
<th>Level</th>
<th>SAE</th>
<th>NHTSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Automation</td>
<td>No Automation</td>
</tr>
<tr>
<td>1</td>
<td>Driver Assistance</td>
<td>Function-specific Automation</td>
</tr>
<tr>
<td>2</td>
<td>Partial Automation</td>
<td>Combined Function Automation</td>
</tr>
<tr>
<td>3</td>
<td>Conditional Automation</td>
<td>Limited Self-Driving Automation</td>
</tr>
<tr>
<td>4</td>
<td>High Automation</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Full Automation</td>
<td>Full Self-Driving Automation</td>
</tr>
<tr>
<td>SAE level</td>
<td>Name</td>
<td>Narrative Definition</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0</td>
<td>No Automation</td>
<td>the full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems</td>
</tr>
<tr>
<td>1</td>
<td>Driver Assistance</td>
<td>the driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task</td>
</tr>
<tr>
<td>2</td>
<td>Partial Automation</td>
<td>the driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task</td>
</tr>
<tr>
<td></td>
<td><strong>Automated driving system (&quot;system&quot;) monitors the driving environment</strong></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Conditional Automation</td>
<td>the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene</td>
</tr>
<tr>
<td>4</td>
<td>High Automation</td>
<td>the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene</td>
</tr>
<tr>
<td>5</td>
<td>Full Automation</td>
<td>the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver</td>
</tr>
</tbody>
</table>

Figure 2.1: Details of levels of automation presented by SAE [4]
Since the definition presented by SAE is the most complete one, further discussion will be based on the six levels depicted in Fig. 2.1. More details regarding NHTSA’s definitions can be found in [18]. As shown in the Fig. 2.1, the levels of driving automation can be divided into two big groups depending on whether the automated system or a human driver should monitor the environment. Another interesting aspect described in the SAE definition is the “fallback performance of dynamic driving task”. In other words, it is the definition of who shall be responsible for the “dynamic driving task”, if unexpected situations occur while the automated driving system is active. SAE’s definition of the dynamic driving task includes operational (e.g. accelerate, brake, steering, etc.) and tactical (e.g. making decisions to change lanes, use signals, etc.) driving tasks, but not the strategical driving task (e.g. choosing a route).

CACC applications usually require at least “driver assistance (level 1)” to operate. In level 1, automated system execute either lateral (steering) or longitudinal (acceleration/deceleration) control. Automated longitudinal control is usually the case for CACC as well as most ADAS. Current efforts as reviewed in [19] has been made to push the applications to level 2 and 3.

2.1 Related Works

2.1.1 Modelling and Simulation of C-ITS

Network and Traffic Simulation

Network simulators have been widely used to aid several studies about wireless communication in C-ITS, which normally is referred to as vehicular ad-hoc network (VANET). However, most of the simulators are not tailored for simulating C-ITS scenarios, because the lack of realistic modelling of the communication nodes’ mobility. Adequate modelling of road traffic is required to estimate positions and movements of involved network components. Therefore, realistic mobility of the nodes is essential. Such vehicle mobility models in network simulations can be pre-generated traces, either from recorded vehicles traces in real world, or another simulation tool. For instance, [20] presents MOVE, a realistic trace generation approach for ns-2\(^1\) and Qualnet\(^2\). Major issue with this approach is that the traces are fixed and cannot be changed during simulation. This makes it difficult to study real-time interactions between actors (e.g. vehicles, infrastructures, pedestrians), especially on driver behaviours. Alternatively, a network simulator can be coupled with a traffic simulator to obtain mobility of communication nodes. For example, Simulation of Urban Mobility (SUMO) coupled with the ns-3 network simulator, and with Veins, as presented in [13] and [12] respectively. Summary of different approach on mobility modelling for VANET is provided in [21].

\(^1\)http://www.isi.edu/nsnam/ns/ (accessed 2 August 2016)

\(^2\)http://web.scalable-networks.com/content/qualnet (accessed: 2 August 2016)
Traffic simulation can be seen from two points of view: macroscopic and microscopic. The macroscopic simulation models the traffic as flows with relationship to the traffic density and speed in a section of the transportation system, such as a highway. As opposed to macroscopic, microscopic traffic simulators are often used for C-ITS simulation. The microscopic approach models movements of each vehicle individually using car-following and lane-changing models. Although traffic simulation alone might be able to assess some aspects of C-ITS (as presented in [22]), several traffic simulators have been used in combination with network simulators in studies related to C-ITS. For instance, AIMSUN [23] in [24], Paramic [25] in [26, 27], VISSIM [28] in [29], and SUMO [30] in [31, 13, 12, 32]. An exhaustive list of simulators for vehicular ad-hoc networks (VANETs) is presented in [33, 34].

Driving Simulation

Driving simulators have been used in many research areas such as human factors, highway design, vehicle dynamics, etc. There are many different types of driving simulators, as stated in [35], “Depending on the needs of the researcher, simulators have ranged from a simple set of pedals that a driver reacted with when a light turned on, to entire facilities dedicated to creating the most realistic simulator by using actual car cabs strapped to moving platforms.” Nonetheless, they all serve the same purpose, that is to obtain measures of driver and driving performance in repeatable and controlled driving environment.

During the transition to a cooperative automated driving era, human drivers will still be involved in the driving tasks such as monitoring the vehicle, interacting with other systems through ADAS, etc. Therefore, involving the human driver in the studies is essential for design and development of future C-ITS applications. Therefore, driving simulators are needed, but the traditional driving simulators are not capable to perform C-ITS studies on their own. Additional capabilities are necessary, such as more detailed and complex mobility and wireless communication models. Driving simulators usually put more focus on driving experience than traffic modelling, and normally do not consider realistic wireless communication models. Thus, driving simulators have been integrated with traffic and/or network simulators to enhance their capabilities in C-ITS simulation as presented in [36, 37, 38].

### 2.1.2 State of the art

#### Cooperative Intelligent Transport Systems

In 2009, European Commission has issued a mandate, M/453 with the title “M/453 STANDARDISATION MANDATE ADDRESSED TO CEN, CENELEC AND ETSI IN THE FIELD OF INFORMATION AND COMMUNICATION TECHNOLOGIES TO SUPPORT THE INTEROPERABILITY
2.1. RELATED WORKS

OF CO-OPERATIVE SYSTEMS FOR INTELLIGENT TRANSPORT IN THE EUROPEAN COMMUNITY". The mandate requested the European Telecommunications Standards Institute (ETSI), European Committee for Standardization (CEN), and European Committee for Electrotechnical Standardization (CENELEC), to identify a coherent set of standards, specifications, and guidelines for implementation and deployment of C-ITS in Europe. During 2000s, several projects focused on vehicular communication infrastructures, technologies, and applications as summarized in [39]. Three main European projects during the period are: CVIS, SAFESPOT, and COOPERS, as mentioned in [40]. Consequently, the European Telecommunications Standards Institute (ETSI) has released the first set of C-ITS standards [41].

Recently, more studies have been focused on development and deployment of C-ITS applications. For example, the Platform for the Deployment of Cooperative Intelligent Transport Systems in the European Union (C-ITS Platform), the European project started in 2014, has published its final report in January 2016. Two main focuses of the project are on technical (e.g. hybrid communication, cyber-security, in-vehicle data access), and legal (e.g. privacy, liability) issues of C-ITS. Apart from the main focuses, topics such as business models, standardisation, public acceptance, etc. are also covered. The goals of the project are: a) establish agreements on how to ensure interoperability of C-ITS; and b) identify most likely and suitable services to be deployed across the European Union (EU). The “master plan” for the deployment of C-ITS will be prepared by the European Commission based on results from the C-ITS platform. In addition, a joint development project between Austria, Germany, and the Netherlands, the Cooperative ITS Corridor project, chose a highway from Rotterdam to Vienna via Frankfurt for implementing two first-step C-ITS applications: road works warning (RWW); and Vehicle Data for improved traffic management. Roadside facilities will be implemented to accommodate the applications, and common conventions will be defined to ensure harmonized interface with vehicles in the three countries. Furthermore, AutoNet2030 is another ongoing European project dealing with development and testing of decentralized decision-making cooperative automated driving technology.

Besides Europe, there are also numerous interests and research related to C-ITS applications, as summarized in [42, 43]. Especially in the United States of America (USA), Japan, and South Korea, as summarized in [44]. Furthermore, the USA has included connected vehicles in the 2015-2019 strategic plan of the United States Department of Transportation (USDOT). California Partners for Advanced Transportation Technology (PATH) is another research and develop-

4 http://www.c-its-korridor.de/ (accessed: 2 August 2016)
6 http://www.its.dot.gov/strategicplan/ (accessed 5 August 2016)
7 http://www.path.berkeley.edu/ (Accessed 5 August 2016)
opment program that has been conducting studies related to C-ITS, especially CACC and truck platooning. China also has ongoing work related to C-ITS such as [45].

**Testing and Evaluation of C-ITS**

In early development phases of C-ITS, simulation tools are used for testing and evaluation, due to various aforementioned reasons. For instance, simulation is cost efficient, repeatable, safe, and does not require real hardware that may not yet exist. Eventually, when the hardware is available, hardware-in-the-loop simulation can be used to test the hardware in simulated environments. Lastly, field operational tests (FOT) may be carried out to assess the system in real operating conditions. A similar approach is presented in [46], with examples based on products from Tass International. Although FOT is the main focus, [47] proposed test architecture including three test environments: simulation environment; test bench environment; and FOT environment.

As aforementioned, recent simulation platforms are usually based on a combination of traffic and network simulators. For example, iTETRIS [13] is a simulation platform for evaluation of C-ITS applications. Focusing on large-scale simulation, SUMO is used as the traffic simulator and ns-3 as the network simulator. The central block, iCS (iTETRIS Control System), handles interaction between C-ITS application, SUMO, and ns-3. The platform architecture is based on communication architecture proposed by ETSI. Similarly, [48] also use SUMO and ns-3 in their approach. In this work, high-resolution modelling of the ego vehicle is emphasized, with VIRES Virtual Test Drive (VTD) modelling driver behaviour, vehicle dynamics, and sensors. Moreover, the test applications are running on the virtual Electronic Control Units (ECUs).

Furthermore, a driving simulator is combined with traffic and network simulators in the work from Zhao et al. [37]. The work presents the integrated traffic-driving-network simulator (ITDNS), that consists of the University at Buffalo driving simulator, PARAMIC (traffic simulator), and ns-2 (network simulator). ITDNS is used to evaluate an “eco-signal” application, a speed advisor application for fuel saving, in two experiments. First, to evaluate the fuel and emission saving results for human drivers driving with, and without the eco-signal application. Second, to compare the fuel and emission saving results between human driver and fully autonomous vehicles, in relation to the speed profile (acceleration/de-acceleration) of the vehicles. To the author’s knowledge, this is one of the first work that uses a combination of driving, traffic, and network simulation to evaluate C-ITS applications.

Apart from the “simulators combination” approach, [49] presented an extension to a traffic simulator, MovSim [50], using multi-agent simulation approach.

---

8https://www.tassinternational.com/ (Accessed 5 August 2016)
2.2 Background

In addition to mobility of vehicles in the system, wireless communication plays an important role in C-ITS. A sufficient modelling of C-ITS needs to model mobility and wireless communication with a certain accuracy. Among the aforementioned simulation tools, Plexe [32] is chosen because of the author’s interest in CACC applications and its availability as an open source tool. Plexe is the platooning extension for Veins [12]. Veins is an open source vehicular network simulation framework, which based its models execution on an event driven network simulator, OMNeT++ [51]. To obtain mobility of the communication nodes in simulation, Veins interacts with SUMO (simulation of urban mobility [30]) via the traffic control interface (TraCI) [52]. Plexe made number of extensions and modifications to SUMO and Veins, the Plexe’s version of them will be referred to as plexe-sumo, and plexe-veins respectively, in this thesis.

In plexe-sumo, models of CC, ACC, and two CACC controllers are available as car-following models. The implemented CACC controllers are from the work by Ploeg et al. [53], and Rajamani [54, Chapter 7]. Moreover, a first order low-pass filter from [54, Chapter 5] is implemented to model the power train behaviour of the vehicles, as illustrated in Fig. 2.2. Equation 2.1 elaborate the computation of acceleration at the simulation step \( n \), \( \ddot{x}[n] \). Desired acceleration is defined as \( \ddot{x}_{\text{des}} \), and \( \beta \) is computed from a constant \( \tau \) and the time step \( \Delta t \) as shown in equation 2.2. In this thesis, 0.5 and 0.01 second are used respectively for \( \tau \) and \( \Delta t \). Thus, one simulation step \( n \) is equivalent to \( \Delta t = 0.01 \) second.

\[
\ddot{x}[n] = \beta \cdot \ddot{x}_{\text{des}}[n] + (1 - \beta) \cdot \ddot{x}[n - 1] \tag{2.1}
\]

\[
\beta = \frac{\Delta t}{\tau - \Delta t} \tag{2.2}
\]

Table 2.2: Network parameters in plexe-veins

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path loss model</td>
<td>Free space (( \alpha = 2.0 ))</td>
</tr>
<tr>
<td>PHY model</td>
<td>IEEE 802.11p</td>
</tr>
<tr>
<td>MAC model</td>
<td>1609.4 single channel (CCH)</td>
</tr>
<tr>
<td>Frequency</td>
<td>5.89 GHz</td>
</tr>
<tr>
<td>Bitrate</td>
<td>6 Mbit/s (QPSK R = ( \frac{1}{2} ))</td>
</tr>
<tr>
<td>Access category</td>
<td>AC_VI</td>
</tr>
<tr>
<td>MSDU size</td>
<td>200B</td>
</tr>
<tr>
<td>Transmit power</td>
<td>20 dBm</td>
</tr>
</tbody>
</table>

For plexe-veins, TraCI interfaces are modified to interact with new car-following models such as passing parameters to CACC, obtain current controller settings, etc. Each vehicle is now equipped with a basic network stack.
including IEEE 802.11p network interface card, basic message dissemination protocol, and an application layer. An example of a platooning scenario with one platoon of vehicles is also provided with the original release. Network parameters from the example is used in this work, as listed in Table 2.2.

Figure 2.2: The first order low-pass filter applied to the vehicles in plexe-sumo.

Figure 2.3: The “Sim IV”, driving simulator with motion system at VTI
Inspired by the work from Zhao et al. [37], the driving simulation software from the Swedish National Road and Transport Research Institute (VTI) is connected to Plexe. VTI’s driving simulation software is developed in house at VTI, it mainly consists of C++ components. There are three main modules; VISIR - for graphic rendering, SIREN - for the sound, and CORE - the kernel software running the main simulation loop. CORE also include vehicle dynamic models, scenario description, cabin interface, and human-machine interface (HMI) software. The driving simulation software can either run on desktop environment, or simulators with a physical motion system at VTI as illustrated in Fig. 2.3. In this thesis, the driving simulation software only run on a desktop computer.

The C-ITS simulation framework presented in this thesis, is made by connecting the VTI’s driving simulation software with the Plexe framework for CACC evaluation. Approaches taken, challenges, as well as results, will be presented in the next chapter, Chapter 3.
Chapter 3
Contributions

C-ITS involves a wide range of research areas such as human factors, communication, vehicle dynamics, software engineering, etc. Paper I (Dimensions of Cooperative Driving, ITS, and Automation) attempts to capture the important complexity influencing aspects of C-ITS. The paper also defines and summarizes C-ITS in relation to driving automation. It further discusses C-ITS from the driver behaviours, and software structure perspectives. Lastly, it discusses challenges related to C-ITS, which include testing and evaluation of C-ITS. Consequently, two interoperability issues are identified from the analysis of challenges in C-ITS: a) interoperability between vehicles with different capabilities, e.g. not operating on the same level of automation; and b) interoperability of same C-ITS function from different vendors, whether they will be able to operate together. These two are important issues that have not been frequently studied.

Paper II and III contributes to the main research question of this thesis (How to create a simulation environment for CACC evaluation?) Paper II proposed an extension for the driving simulation software from the Swedish National Road and Transport Research Institute (VTI), to include models illustrated in Fig. 1.1. Traditional driving simulators normally do not consider V2V communication. Moreover, modelling of the surrounding traffic in driving simulators is often simplified and has limited scope, i.e. only in the area around the ego vehicle. Therefore, the driving simulation software from VTI is extended with an existing traffic and network simulation framework for platooning applications, namely Plexe. The proposed extended driving simulator covers most of the models in Fig. 1.1, only vehicle dynamics and human machine interface (HMI) are omitted. The proposed simulation framework is illustrated in Fig. 3.1, omitted models are marked with black boxes.

The extended driving simulator enables possibilities to study CACC with human driver in the loop from many aspects using one tool. For example, effects of failure in communication on human drivers. Furthermore, one of the presented use cases presents the potential to control a vehicle in the simulation...
with an external source. The use case elaborate on the possibility of having several different external sources controlling vehicles in the simulation. Lastly, remaining challenges and limitations of the proposed driving simulator are discussed in the paper. One of the limitations, which is later solved and presented in Paper III is how to model and handle the lane changing manoeuvre in the simulator, since the simulation of urban mobility (SUMO) (the traffic simulator used in Plexe) does not consider lateral acceleration. The lane changing in SUMO occur in one time-step, i.e. a vehicle switch from one lane to another instantaneously.

Paper III develops the extended driving simulator further. First, a more realistic lane changing manoeuvre is implemented in the VTI’s driving simulator. Vehicles are no longer changing lane instantaneously in the driving simulator. However, the solution has only effect on the visualization. Vehicles in SUMO are still changing lane instantaneously. Second, a simplified version of the platoon merging scenario from the GCDC 2016 [55] is implemented and evaluated. The interaction protocol and communication message set for platoon merging is also implemented in the simulation framework. The scenario presents more use cases, and illustrates some of the potentials of the proposed C-ITS simulation framework with current level of abstraction in the models.

### 3.1 Summary of Paper I

This paper presented a definition of C-ITS, analysis from driving behaviour and platform architecture perspectives, and challenges in C-ITS as well as integrating driving automation into C-ITS.

Automated driving can be classified by “levels of driving automation” proposed by organizations such as SAE, BASt, NHTSA, and VDA. Table 2.1 presents levels of driving automation proposed by SAE and NHTSA. Each level of driv-
ing automation are related to the complexity of the systems by defining the limits of automated tasks. For instance, at the level 2, the driver has full responsibility to monitor the surroundings, and the vehicle can take control of itself in some driving modes, e.g. while ACC and LKA are turned on at the same time. With the complexity related to the levels of driving automation, developers know which requirements needed to be fulfilled.

On the other hand, C-ITS has not been as clearly defined as automated driving has. Interactions or negotiations between vehicles and infrastructures play an important role in C-ITS. Reliable communication is one of the requirements towards successful interactions. As a result, the complexity of C-ITS is expected to be higher.

Therefore, three dimensions of C-ITS are presented: a) the number of actors in the system; b) the driving tasks; and c) the scope of goals. They are illustrated in Fig. 3.2. The complexity of C-ITS grow from the origin outwards.

![Figure 3.2: Dimensions of cooperative ITS](image)

Actors in the system are vehicles and infrastructure objects. Starting from two, adding more actors to the system will increase the complexity. C-ITS functions need to handle more interactions and consider cases when it fails to communicate or the other actors do not cooperate.

Further, according to the driver behaviour models in [56], driving can be seen as problem solving tasks: operational, tactical, and strategical. Tasks such as steering and pushing accelerate/brake pedals are seen as operational. Operational tasks are usually less complex than the others. Tactical tasks involves short-term decision making, e.g. whether to change lane, cross intersection, etc.
Strategical is planning of the whole journey. For instance, route choices, driving goal, e.g. save fuel, reach destination as fast as possible, etc. Strategical tasks require a lot of information, and are normally more complex among the three tasks. Therefore, depending on which driving task it is solving, the complexity of a C-ITS function will be different. This can also depend on the purposes of the function. For example, functions with safety as the main goal might be more complex than the ones aiming at driver’s comfort.

Lastly, the scope of goals is another factor that can affect complexity of C-ITS. There are three levels of the scope: individual, local, and global. A C-ITS function may have more than one goal. The scope of goals in this context means the scope of actors that would benefit from the function reaching its goals. For instance, making way for an emergency vehicle is a good example of individual benefits. Actors in the system cooperate to give benefits to one vehicle. Local scope refer to a small area such as an intersection, a highway exit, etc. And the global scope will give benefits to actors in a city area or the whole region.

Challenges towards deployment of C-ITS are also presented in the paper. First, providing sufficient communication coverage with reliability is one big challenge to be solved. Second, interoperability issues need to be considered. Most of the research are done with the assumption that vehicles are identical or capable of operating at the same level of driving automation. However, there will be a mix of different vehicles in real driving situations, which need to cooperate. Third, challenges regarding safety of the C-ITS functions. International Organization for Standardization (ISO) released ISO 26262 [57], which defines a functional safety standard for automotive electrical and/or electronics systems. A procedure for hazard analysis and risk assessment, which result in Automotive Safety Integrity Level (ASIL) is proposed. However, it does not cover systems that involves V2X communication and there is no other such standard defined for C-ITS. Last but not least, C-ITS introduce more complex scenarios and new possibilities, hence going through all of them is almost impossible. Therefore, a new methodology might be required to ensure that sufficient testing and evaluation has been done.

### 3.2 Summary of Paper II

This is a short paper describing initial works with developing an extended driving simulator aimed for evaluation of C-ITS. Figure 3.1 presents components to be modelled in C-ITS simulation. Driving simulators normally have limited capability for modelling of communication and surrounding actors. Therefore, an extended driving simulator framework has, in this paper, been proposed for C-ITS evaluation. Two major motivations are: a) to model V2V communication, which usually is not available in driving simulators; and b) improve models of surrounding vehicles.

The paper presents the extension of the driving simulation software from the Swedish National Road and Transport Research Institute (VTI). The driv-
3.2. SUMMARY OF PAPER II

The driving simulator is extended with a network and traffic simulator, Veins [12] and Simulation of Urban Mobility (SUMO) [30] respectively. Moreover, the Plexe version of SUMO and Veins are used in the paper (plexve-sumo and plexve-veins). The extension uses two transmission control protocol (TCP) connections, TCP\textsubscript{app} and TCP\textsubscript{sync} as shown in Fig. 3.3. TCP\textsubscript{app} handles data exchanges between plexve-veins and the driving simulator. TCP\textsubscript{sync} handles synchronization between Plexe and the driving simulator. PLEXE-VEINS and plexve-sumo are connected in client-server fashion, with plexve-sumo as a server. Exchange of data and synchronization is done through traffic control interface (TraCI) [52] protocol over a TCP connection.

With plexve-sumo as a server, at each update interval, which is usually one time step in plexve-veins, it request plexve-sumo to execute until target simulation time. The driving simulator synchronizes with plexve-veins in a similar way. Hence, at each update interval in plexve-veins, it waits for a synchronization message from the driving simulator. Apart from handling the synchronization, TCP\textsubscript{sync} is also used to forward all vehicle parameters, that plexve-veins subscribed, to the driving simulator. The parameters include vehicle’s name, speed, and positions (x and y) at synchronized time points.

All simulators are running at 100Hz (0.01 second time step). A scenario with one platoon of five vehicles is simulated for 120 seconds. During the simulation, the lead vehicle slow down and speed up, at simulation time 40 and 100 seconds respectively. Moreover, at 60 seconds simulation time, each vehicle in the simulation increase its desired distance to the preceding vehicle.

As results, two use cases are presented. The first use case elaborates on the possibility to use the extended driving simulator framework to study human factors within C-ITS, by visualizing behaviour of two CACC controllers in the VTI’s driving simulator in real-time. The existing CACC controllers in plexve-sumo are used: a) CACC controller proposed in [54, Chapter 7]; and b) The controller proposed by Ploeg et al. [53].

The second use case illustrates the flexibility of the extended driving simulator. A simple but challenging control logic resulting in a step response shown in Eq. 3.1 is used to compute desired speed for the ego vehicle at the simulation step n (\(\dot{x}_{i,\text{des}}[n]\)). From the Figure 3.4, \(\dot{x}_i\) is the speed of the ego vehicle, \(\dot{x}_{i-1}\)
is the speed of the preceding vehicle, and $\text{gap}_{\text{des}}$ is the desired inter-vehicle distance. In normal case, vehicles in the simulation are controlled by a selected car-following model in *plexe-sumo*. The output of the car-following model is sent through the model of actuator in *plexe-sumo* (the low-pass filter elaborated in Equation 2.1). Using the control login in Eq. 3.1, the desired speed of the ego vehicle, $\dot{x}_{i,\text{des}}$, is computed and sent to *plexe-sumo* to control the vehicle via the actuator model (the low-pass filter elaborated in Equation 2.1). Inter-vehicle distances measured in the driving simulator are used as a reference for $\text{gap}_{\text{des}}$.

\[
\dot{x}_{i,\text{des}}[n] = \begin{cases} 
\dot{x}_{i-1}[n - 1] - 5 \text{km/h}, & \text{if } \text{gap}_{\text{des}} < 12 \text{meters} \\
120 \text{km/h}, & \text{otherwise} 
\end{cases} 
\]

(3.1)

One simulation step $n$ is equal to one time step, $\Delta t = 0.01$ second. The desired speed of the ego vehicle obtained from Eq. 3.1 is translated to the desired acceleration using the Eq. 3.2 below.

\[
\ddot{x}_{i,\text{des}}[n] = \frac{\dot{x}_{i,\text{des}}[n] - \dot{x}_i[n]}{\Delta t} 
\]

(3.2)

The desired acceleration in Eq. 3.2 is then used in the actuator model (the low-pass filter modelling kinematics of the vehicles presented in Section 2.2):

\[
\ddot{x}[n] = \beta \cdot \ddot{x}_{\text{des}}[n] + (1 - \beta) \cdot \ddot{x}[n - 1]
\]

The controller can be executed either in *plexe-veins* or the driving simulator. In case the controller is executed in *plexe-veins*, it communicate with the driving simulator using the TCP app connection to obtain the current distance to the preceding vehicle. On the other hand, when the controller is executed in the driving simulator, *plexe-veins* requests for the resulting speed, which is the $\dot{x}_i$ is calculated in the driving simulator. Lastly, this use case also shows that the simulators are time and space(position) synchronized.

In summary, the paper presented an evaluation of the extended driving simulator with two use cases. The use cases are intended to elaborate on the potential of the simulator. For instance, usage in the area of human factors. Also, they
are used to preliminarily evaluate the simulator itself. Apart from synchronization and exchange of information between simulators, many challenges are still remaining. First, the driving simulator is not yet aware of any V2X messages in *plexe-veins*. Extending it to be aware of the messages would allow more scenarios involving interactions from human drivers. Moreover, traffic simulators such as SUMO do not consider lateral acceleration. Therefore, the lane changing occurs instantaneously, i.e. vehicles switch from one lane to another in one time step. Nevertheless, tactical driving decisions, e.g. when to change lane, can already be evaluated using the proposed simulator.

### 3.3 Summary of Paper III

This paper is the continuation of the **Paper II**. The paper used the same simulation framework presented in the **Paper II** with improvements. It presents a simplified implementation of the platoon merging protocol and the platoon merging scenario from the GCDC 2016 competition in the simulation framework.

The extended driving simulator presented in **Paper II** is developed further in this paper. First, it extends the basic platooning example in *plexe-veins* to have two platoons running in parallel instead of one. Furthermore, the lane changing model for CACC in *plexe-sumo* is changed to ignore the “safe gap check”. The lane changing model will perform the check before making a decision to change lane. If the gap in the other lane is not large enough, the lane change will not happen and the vehicle will try to overtake instead. Since vehicles are driving close to each other in a platoon, the gap is always considered not safe, which results in the vehicle trying to overtake the whole platoon. Therefore, the check is removed to allow freedom in lane change. On the driving simulator side, more realistic visualization of lane change manoeuvres is used. When lane changes happen in *plexe-sumo*, the driving simulator uses an existing lane-changing function to perform the manoeuvres. The manoeuvre is implemented by a proportional-integral-derivative (PID) controller, which control yaw velocities based on errors in lateral position.

Moreover, the platoon merging scenario from GCDC 2016 is implemented in the simulation framework. The GCDC scenario is chosen because of two main motivations: *a*) it is an interesting scenario involving interactions between vehicles and lane changes; and *b*) Halmstad University is participating in the GCDC 2016, where data from competing vehicles are logged and made available and thus, can be used to validate the simulation framework. Therefore, the message set for platoon merging, as defined from the organizer of GCDC 2016, is added to *plexe-veins*. The platoon merging protocol is implemented in *plexe-veins*, and the platoon merging scenario is simulated.

The platoon merging scenario starts with two platoons of vehicles driving in two lanes, one platoon per lane. The platoons receive the “merge requested” message, then they initiate the merging because one of the lane will be closed
due to road maintenance ahead. The vehicles in the platoons communicate and make gaps for each other. Finally, the two platoons merge to one platoon and drive past the construction zone. An overview of the scenario is illustrated in Fig. 3.5

![Figure 3.5: Overview of the simulated platoon merging scenario](image)

A simple gap making strategy is evaluated in this paper. The results show that the simulation framework can simulate and analyse the gap-making strategy. Two scenarios are simulated, which are designed for evaluation of the gap-making strategy with different parameters, and with different CACC controllers.

In conclusion, an improved version of the extended driving simulator is presented. The improvements include the capability to visualize a realistic lane-changing manoeuvre, and the ability to simulate the platoon merging scenario from GCDC 2016. This paper finally also elaborate on potential of the simulation framework and possibility to execute more complex scenarios.
Chapter 4
Conclusions

Modelling and simulation for evaluation of C-ITS functions are discussed in this thesis, in particular, CACC and its applications such as platooning, platoon merging, etc. A simulation framework for evaluation of CACC applications is presented. With the combination of VTI’s driving simulator and Plexe, the simulation framework can be used to addressed many issues regarding evaluation of CACC applications. For example, human factors studies, effects of malfunctions in communication network on traffic system and CACC applications, and mixed traffic scenarios (cooperative and non-cooperative vehicles in the traffic). These examples are challenging scenarios in C-ITS that need to be tested before deploying a C-ITS application. Although they are not studied in this thesis, the simulation framework is capable of studying them, and they are listed as future works.

The first appended publication, Paper I, identifies challenges, scope, and definition of C-ITS. Complexities of C-ITS is discussed and dimensions of C-ITS is presented. These dimensions are intended to express capabilities and complexity of the C-ITS functions. Consequently, researchers can better define the requirements, and test cases based on them. The simulation framework is presented in Paper II and III. A way of modelling C-ITS is presented, as depicted in Fig. 1.1. The simulation framework consists of:

1. VTI’s driving simulation software.
2. plexe-veins, the Plexe version of Veins, vehicular network simulator.
3. plexe-sumo, the Plexe version of SUMO, microscopic traffic simulator.

A few use cases of the simulation framework with CACC applications are presented. The purpose of the use cases are twofold. First, to ensure that the simulation framework is working as intended, they serve as test cases for it. Second, they elaborate potential usages of the simulation framework.

In summary, the current simulation framework has potential in three research directions:
1. human factors, e.g. studies related to effects of CACC applications on human drivers;

2. functional safety aspects of the CACC applications, e.g. resilience of the function to communication malfunctions; and

3. effects of CACC applications on the large-scale transportation system level, e.g. how many connected and cooperative vehicles are needed in order to improve the transportation system.

4.1 Discussion

As presented in Paper II, the proposed simulation framework has potential to control simulated vehicles with external sources. Therefore, the simulation framework can incorporate a model running in an external software such as Simulink, or hardware such as an electronic control unit (ECU). However, interfaces from the simulation framework to the external models need to be provided, preferably with a standard interface such as Functional Mock-up Interface (FMI)\(^1\). Moreover, using a standard architecture for distributed simulation, could increase scalability and flexibility. For example, High-Level Architecture (HLA) as presented in [58, 48].

Paper III only solved the lane changing manoeuvre visually. In SUMO, the vehicles are still changing lane instantaneously, which will have effects on sensors and actuators modelled in SUMO. This issue requires further analysis regarding required abstraction level of the models. Perhaps a C-ITS application such as platoon merging does not require modelling of detailed lane changing manoeuvres in order to be tested and evaluated. Nevertheless, this is one of the important challenges that needs to be addressed in the future works.

The presented simulation framework proposes an approach to answer the main research question of this thesis (How to create a simulation environment for CACC evaluation?). Moreover, the more specific research questions have been tackled. Required models are presented in Fig. 1.1 to elaborate on the RQ 1.1 (What need to be modelled and simulated for testing and evaluation of CACC?). From the author’s point of view, these models are adequate to model CACC applications and complexities of C-ITS, discussed in Paper I.

As stated in RQ 1.2 (What are the level of abstraction and accuracy needed in each model?), having sufficient level of abstraction, or level of detail, in each model is an important factor in simulation studies. Since widely used simulation tools are chosen as basis (SUMO and Veins), many existing models with different levels of abstraction are available. The simulation framework are using the ones that Plexe’s developers have chosen, which are supposed to be suitable for platooning applications. Nevertheless, further investigations may be required

---

\(^1\)https://www.fmi-standard.org/ (Accessed 11 August 2016)
4.1. DISCUSSION

Figure 4.1: Comparison of the speed profile when accelerate from 0 to 30 km/h.

whether accuracy of the model are sufficient. For example, whether vehicle dynamics models are required instead of the low-pass filter used in plexe-sumo, or whether a realistic lane changing manoeuvre are required in SUMO, as afore-
CHAPTER 4. CONCLUSIONS

mentioned. Comparison between logged speed profile of the Halmstad team’s vehicle in GCDC 2016 and the low-pass filter is illustrated in Fig. 4.1. It is difficult to judge whether the low-pass filter is sufficient without doing extensive numerical analysis. Moreover, the speed profile of a vehicle depends on many factors such as engine power, gear, weight of the vehicle, etc. Nevertheless, the Fig. 4.1 depicts the accuracy of low-pass filter used in Plexe, compared to the vehicle in real-world.

Current version of the simulation framework has been assuming no disturbance in wireless communication, positioning systems, and sensor readings. Model of disturbances can also be added in the future, to provide more accurate modelling, such as disturbances in wireless communication, and errors in sensor readings, map, and GPS positions, etc. However, since the driving simulator is involved in the simulation framework, real-time performance is an important requirement. More accuracy of models might require higher computational power, which may degrade the real-time performance of the framework. Therefore, given limited computing resource, balance between accuracy of the models and simulation performance needs to be considered. Nonetheless, the required computational power is also related to the level of abstraction. For instance, at the lower abstraction level, more detail can be added, thus requires more computational power.

Regarding RQ 1.3 (How to integrate and synchronize the simulators?) and RQ 1.4 (What are the interfaces required to incorporate existing models?), TCP/IP connections has been used to connect and synchronize the simulators, with master-slave scheme and TraCI interface as presented in the Paper II. Also, the Paper II provide an example of how to interface between an external controller to the existing vehicle model in SUMO, which is also done via a TCP connection. There are many other approaches to integrate and synchronize the simulators. For instance, using HLA, or central software to interact with other simulators through interfaces such as in iTETRIS [13]. The approach taken in this thesis is more simple than the other two approaches. However, it might not provide great scalability as HLA could provide. Moreover, it may have high dependency on the version of Plexe, if the code structure in Plexe is changed significantly.

On model-level interfaces, standards such as FMI can be used. The standard was released in 2010 to create a tool-independent support for model exchange and co-simulation. Furthermore, to efficiently interface many components, Lightweight Communication Marshalling (LCM) [59] is an alternative. LCM is a low-latency message passing and data marshalling library. It utilizes subscribe/publish message passing scheme with User Datagram Protocol (UDP) multicast as its underlying transport layer. LCM is independent of programming language, and currently support C, C++, C#, Java, Lua, MATLAB, and Python. The two options above could be implemented in the future work, enabling the simulation framework to interact with external models in standardized manner.
4.2 Future Works

As an important part of planned near future work, the simulation framework needs to be validated with logged data from GCDC 2016 competition, which took place at the end of May 2016. The logged data can be used to validate the simulation framework and assess whether the current level of abstraction in each model are feasible and suitable in relation to the real situations. The evaluation process will be able to conclude on how sufficient the proposed framework is. Hence, the evaluation results will be used to direct and guide the future development of the simulation framework. Moreover, Plexe has release its new version (version 2.0). If the decision is to continue using Plexe, an upgrade to the newer version will be required.

Making the driving simulation software to be aware of V2V communication messages is an essential task. This will enable interactions between the human driver and the application through an HMI, e.g. change speed of the vehicles, re-route, switch lanes, etc. Consequently, HMI solutions can be evaluate using the simulation framework.

On the “day one” deployment of CACC applications, mixed traffic scenarios between cooperative and non-cooperative vehicles are to be expected. Therefore, to enable a human driver to manually control the ego vehicle is another important step to be considered. The control is not necessarily fully manual, for instance, the human driver can control the lateral position while the CACC executes longitudinal control. This capability would enable human factors studies. For example, studies about driver behaviours and decisions when he/she encounter a platoon of vehicles controlled by CACC.

Furthermore, future research questions such as how can the simulation framework be used to ensure functional safety of CACC applications? need to be answered. Effects of disturbances, such as failures in wireless communication, and errors in GPS and sensors reading, on the safety performance of CACC applications has to be determined. Apart from functional safety, interoperability is also an important challenge in C-ITS. It can be divided into three categories; interoperability between different a) car manufacturer, b) software versions, and c) automated capabilities, i.e. levels of automation. To address these challenges using the simulation framework, processes or methodologies need to be developed.

Last but not least, even though CACC applications are the focus at the moment, a long-term future plan is to aim towards a methodology for development and evaluation of other C-ITS applications using the proposed simulation framework.
References


[34] Aamir Hassan. VANET Simulation, 2009. (Cited on page 8.)


Appendix A

Paper I
Dimensions of Cooperative Driving, ITS and Automation

Maytheewat Aramrattana, Tony Larsson, Jonas Jansson, and Cristofer Englund

2015 IEEE Intelligent Vehicles Symposium (IV)
Seoul, South Korea, June 28 - July 1, 2015
Dimensions of Cooperative Driving, ITS and Automation

Maytheewat Aramrattana¹,², Tony Larsson¹, Jonas Jansson² and Cristofer Englund¹,³

Abstract—Wireless technology supporting vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) communication, allow vehicles and infrastructures to exchange information, and cooperate. Cooperation among the actors in an intelligent transport system (ITS) can introduce several benefits, for instance, increase safety, comfort, efficiency. Automation has also evolved in vehicle control and active safety functions. Combining cooperation and automation would enable more advanced functions such as automated highway merge and negotiating right-of-way in a cooperative intersection. However, the combination has influences on the structure of the overall transport systems as well as on its behaviour. In order to provide a common understanding of such systems, this paper presents an analysis of cooperative ITS (C-ITS) with regard to dimensions of cooperation. It also presents possible influence on driving behaviour and challenges in deployment and automation of C-ITS.

I. INTRODUCTION

With its potential benefits to the transport systems as presented in [1], cooperative intelligent transport system (C-ITS) have recently received a lot of attention. For example, in Europe three large projects that have been dealing with cooperative systems are CVIS, SAFESPOP, and COOPERS. Cooperative Vehicle-Infrastructure Systems (CVIS) [2] focused on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication issues. While, SAFESPOP [3] aimed to enhance road safety via a “safety margin assistant” concept, which detects critical situations in advance, and the use of a “local dynamic map”. COOPERS [4] was focused towards providing safety related real-time information and cooperative traffic management through infrastructure-to-vehicle (I2V) communication. A comparative study of these projects is presented in [5], SARTRE [6], another European project dealt with platooning applications, through the concept of increased “driver comfort”. Five years after the previous competition in 2011 [7], the grand cooperative driving challenge (GCDC) 2016 will be arranged by the i-GAME project [8]. The objective is to speed up real-life implementation of automated driving and interoperability of wireless communication. Besides i-GAME, another ongoing project is AutoNet2030 [9], working towards cooperative automated driving technology based on a distributed decision-making strategy.

[10] present an architecture for cooperative driving of automated vehicles. It consists of three layers a) vehicle control layer; b) vehicle management layer; and c) traffic control layer. The vehicle control layer typically is individual for each vehicle. It is connected to sensing and actuating systems, it sends data from sensors and vehicle state variables to the vehicle management layer. It also receives steering and vehicle speed commands from the vehicle management layer. The vehicle management layer is also implemented in the vehicles, placed in the middle between the vehicle control and the traffic control layer. It determines the movement of each vehicle in the C-ITS, with the data from the vehicle control layer of neighbouring vehicles through V2V communication. It also receives information from the traffic control layer via V2I communication. The traffic control layer consists of two parts; physical and logical. The physical part is located in the infrastructure, it consists of physical equipment like traffic signals, communication access and relay nodes, and roadside units. The logical part deals with regulations, rules, manners, common sense, and ethics in the human society. Considering the two parts, common criteria must be defined and communicated to neighbouring vehicles through the vehicle management layer.

Within C-ITS, information is shared between many actors such as vehicles, infrastructures, cloud services, etc. However, only sharing information is not enough to be considered a C-ITS, cooperation and interaction between the actors in the system is also required. In order to have a common understanding of what we mean by C-ITS, this paper presents an analysis of the topic from different perspectives in Section III. Introduction to driving automation is presented in Section II. Section IV elaborates on dimensions of C-ITS followed by its deployment challenges in Section V. Finally, Section VI conclude the paper.

II. LEVELS OF AUTOMATION

Recent research have focused on automated driving functions like adaptive cruise control (ACC), automated parking, etc. Levels of driving automation have also been defined by organizations such as SAE, BASi, NHTSA, and VDA. A comparison of these definitions is presented in Table I. Apart from ongoing research on automation functions like adaptive cruise control (ACC) and automated parking, several papers on cooperative systems in relation to automated driving concepts are published, see [10]–[13].

The following description will use SAE’s definition as the basis. From level 0 to 2, the human driver has responsibility to monitor the environment. At level 2, the vehicle can take over steering, acceleration and deceleration in some driving modes. At level 3 and 4, the vehicle will monitor the environment, but only for some driving tasks. The differences
between 3 and 4 is the fall-back performance, in other words, who is responsible when the system fails. At level 3, the system still expect the human driver to handle the failure with a request to intervene. On the other hand, the vehicle will handle itself at level 4, for instance, when a failure occurs, the automation system still has to safely handle the vehicle. A request to the human driver may be made at this level, but if the driver does not respond, the system should be able to handle the situation. At full automation, level 5, the vehicle will handle all the driving responsibilities or tasks, including monitoring of the environment.

### III. COOPERATIVE ITS

In this paper, the scope of C-ITS encompass technical systems that applies to actors in the road transport system. Within this scope, C-ITS is defined according to the following definition:

**Definition 1:** C-ITS is a technical system that implements cooperative behaviour based on communication between two or more actors in the system.

Cooperative behaviour is in turn defined as:

**Definition 2:** A cooperative behaviour includes two or more actors working towards a common or mutually beneficial goal, purpose, or benefit; enabled by interaction and information exchange between the actors.

Cooperative behaviour involves actions such as sharing information, taking turns, following instructions from others, etc. Typical goals within the transport system context are, the improvement of safety and increased transport efficiency. When combined with driving automation, having more comfortable driving is another goal of C-ITS. The overall goal is to drive beyond the capability of a human driver or an autonomous vehicle. Thus, comfort as well as safety and efficiency are important goals. However, not every cooperative function must deal with all these goals. For example, a function like cooperative adaptive cruise control (CACC) can improve efficiency of the individual vehicles, but to be more efficient, vehicles could also drive closer to each other to reduce air resistance. This could however increase the risk of accident i.e. different goals can be in conflict with each other and may require different cooperative behaviours. Thus, applying the concept to the transport system needs to be considered carefully at different levels.

Apart from the exchange of system state information, interaction about intentions, planned behaviours, and agreements play an important role in C-ITS. In [14] cooperative driving is defined from a human-machine cooperation perspective, focused on the interaction between a vehicle and its human driver. They proposed five levels of cooperation for human-machine interaction. Those five levels deal with: a) intention; b) mode of cooperation; c) dynamic task and action allocation; d) the human-machine interface; and e) the contact between human and machine. Four out of the five levels were presented in the paper with an example of cooperative lane change scenario. Further evaluation of the concept was presented in [15].

### A. Behavioural Perspectives

A critical review of different driver behaviour models is presented in [16]. According to the article, there are three levels of skill and control in driving, seen as a problem solving task: strategical, tactical, and operational. These three tasks relate to the driver’s decision making and is often mentioned as basis for modelling of driver behaviour. The strategical level can be seen as a planning task, it involves things like cost and risk evaluation, route choice, trip goals. The tactical level is about deciding manoeuvres such as: overtaking, turning, gap adjustment based on the criteria made on strategic level. Moreover, negotiation is also involved at this level, for instance when making decision to cross an intersection, as well as monitoring of traffic since it is the basis for making decisions at this level. Lastly, the operational level handles more continuous and periodic routine tasks such as longitudinal and lateral control, based on environmental input. These are principles that any model of driver behaviour should take into account. Furthermore, information flow, switching and interaction between levels should also be considered. To bring this concept into C-ITS, the goal of cooperative functions could be on different levels but cooperative partners should have common goals on those levels to enable efficient cooperative behaviour.

As elaborated in [14] and [15], human-machine interaction is important for cooperative systems as long as the human driver is still involved in the driving task. Furthermore, at the early stage of C-ITS deployment, some vehicles might not have any communication and automation capabilities at all, some might have automation but not communication, and just a few would have both. How to communicate the intention between those three differently equipped categories of vehicles? How would the driving behaviour of autonomous vehicles be perceived and processed by the human driver in a manually driven car without communication? And vice versa. Those are important question from behavioural perspectives that needs to be addressed.

[17] investigated the effects of automation on tactical driving behaviour, depending on the trust in the system. Most
driving automation today works at the operational level, in which the function (when allowed to take over from the driver) handles longitudinal and lateral control, for example, ACC or lane keeping assist functions. On the tactical level, automation can be involved, e.g. in self-parking systems, but usually the vehicle only provides information to help the driver make decisions about driving tasks. Navigation systems are mentioned in [17] as one example of a function aiding strategic tasks, still it does not take control of the vehicle. Within C-ITS, automation of tactical tasks such as crossing of intersections is possible as presented in [11], [18], [19]. Furthermore, [20] elaborate on the possibility of having automation at the strategic or tactical level.

### B. Structural Perspectives

From a structural perspective, actors in C-ITS consists of components aimed for: a) communication; b) sensor fusion; c) environment perception; d) decision making; e) actuators; and f) human driver interaction. Figure 1 illustrate these components from structural perspectives in relation to the driving tasks presented in section III-A.

**Fig. 1: Vehicle/infrastructure actors internal structure.**

Access to one or more mechanisms for wireless communication is one of the key factors that enable C-ITS. Reliable and standardized communication techniques providing sufficient coverage and quality of service in different environments is an important enabler of C-ITS, and may eventually be achieved through a combination of vehicle-to-vehicle and vehicle-to-infrastructure (V2X) and cellular communication systems. ETSI [21], The European telecommunication standard institute, has published two technical specifications [22],[23], defining two types of messages namely cooperative awareness message (CAM) and decentralized environment notification message (DENM) respectively. These two message types are intended for the European C-ITS applications. CAM periodically provide information of presence, position, etc. In case the vehicles maintain a local map of the surroundings, another goal is to locate itself in the environment through these sensors. CAM

**Communication**

**Operational**

**Tactical**

**Strategical**

**Environment perception**

**Sensor fusion**

**Decision making**

**Actuators**

**HMI**

**Human**

**Fig. 1: Vehicle/infrastructure actors internal structure.**

**Operational**

**Tactical**

**Strategical**

**Environment perception**

**Sensor fusion**

**Decision making**

**Actuators**

**HMI**

**Human**

Decision making is an important part of C-ITS to select upon strategy and tactics of the systems based on the information gathered via sensors and communication. C-ITS can have either centralized decision making parts placed in the infrastructure or in a vehicle responsible for a group of vehicles. Alternatively, the decision making could be distributed and decentralized among vehicles and infrastructure. In a complex system both could be used at the same time in combinations such as distributed over the country but centralized within local areas. Depending on the strategy and tactics of the systems, decision making can be divided into short-term, mid-term, and long-term decision making. For instance, short-term decisions are, e.g. manoeuvres for collision avoidance, lane change, etc. They usually need information from the communication module in real-time, otherwise it could be dangerous to the system. For example, the driver receive the notification about a manoeuvre too late, and could not react in time, which might lead to an accident. On the other hand, route choice of a trip is an example of long-term strategic decision making.

At the highest level of automation, actuators, i.e. throttle, brake, and steering wheel, would be totally controlled by the system. However, at the lower level of automation, the human driver is still involved and have effects in this part as well. Especially at levels that are partly automated, the human drivers will have interactions with the decision making part via human-machine interface (HMI) possibly including hap-
tic feedback by force on steering wheel. Moreover, according to the “convention on road traffic” from Vienna 1968 [25], which aim to set up international uniform traffic rules, “every driver shall at all times be able to control his vehicle or to guide his animals”. Thus, if the future policy will follow this rule, the human driver shall always have priority to decide and override the manoeuvre decided by the system.

Lastly, the human driver, interact with the system through its HMI. The human driver, responsible for all driving tasks, has the highest priority to decide and override the decisions from the automation system. Still, some systems such as advanced emergency braking system (AEBS), or anti-lock braking system (ABS), the system override the human driver’s decision. By interactions between the human driver and the system through HMI, the driver will understand the intention of the system and vice versa. Moreover, having access to the communication part allows the driver to make requests to cooperative partners as well as to respond to requests. For example, as elaborated in [14], in the cooperative lane change scenario, although the request is initiated by the software function in a vehicle, the driver in another vehicle could decide, and confirm through HMI to the first vehicle that the request is accepted.

IV. DIMENSIONS OF COOPERATIVE ITS

![Diagram: Cooperative ITS with two and four actors](image)

Fig. 2: Cooperative ITS with two and four actors respectively, interactions are indicated with red dotted lines.

Cooperation between two actors in C-ITS is illustrated in Fig. 2a with the red dotted lines representing possible interactions. The hardware box includes sensors, actuators, communication devices, computers, and user interfaces. There is usually at least one vehicle among the actors, and the other could be another vehicle or a road side unit. In case of cooperation between a vehicle and infrastructure, there is no interaction between the human operators. Cooperation between the vehicle and the infrastructure is usually aimed to assist the driver of the vehicle by providing extra information. In other words, it is typically a one-way communication from the infrastructure to the vehicle. Examples of use cases defined in [26] are: speed limit notification, traffic condition warning, point of interest notification, etc. The next step is when both actors are vehicles. This step includes more advanced scenarios such as cooperative lane change, motorway merging, intersection crossing, etc. Once the number of actors increase, the systems become more complex as illustrated in Fig. 2b. Moreover, if the vehicles are operating at different levels of automation, the interaction become even more complex.

The communication, which enables interaction and leads to cooperation, can be divided into three levels: a) human interaction; b) one-way wireless communication; and c) two-way wireless communication. Today, interaction between human drivers by means of conveying vehicle behaviour is performed via turn signal, vehicle horn, vehicle direction and position, etc. To enable interaction and communication within the transport system, drivers combine vehicle behaviour with eye contact and body language. Moreover, FM-radio sometimes acts as a road side unit providing warnings regarding traffic information. However, as of today, none of the above can communicate with automated vehicles. Thus, the one-way and two-way wireless communication provide channels to interact with and between automated vehicles. Normally, in one-way communication the warning would be sent to the vehicles and it depends on the driver or the automation system to react to the information. Hence, the action rely on the driving behaviour of the driver or the system. With two-way communication, more interaction such as acknowledgement and negotiation is possible. Therefore, it would be able to utilize the benefits of C-ITS.

Although complexity of C-ITS can be defined by many different factors as mentioned above, in this paper, the three dimensions considered important are: a) the number of cooperative actors; b) the driving task (planning horizon); and c) the scope of cooperative benefits.

Starting from two actors, which is the basis of C-ITS, adding more actors to the system will result in a more complex system as illustrated in Fig. 2b. Interaction between actors in the C-ITS is typically realized through wireless communication. More actors will require more reliable communication, more bandwidth and maybe even broader communication range. Furthermore, handling uncertainties created by the actors will also be an issue.

The cooperative function and its interaction behaviour also influence the complexity, depending on which kind of driving tasks it solves, i.e. operational, tactical or strategical. Moreover, depending on the function, different type of interaction is required. For example, CACC operates at the operational level and once the platoon is set up, no interaction between drivers is required unless there is a failure. On the other hand, cooperative lane change, which operates at the tactical level, requires interaction between vehicles and drivers at many different stages e.g. making lane change request, the driver
determines situation and accept/reject the request, etc.

Another perspective is to classify the function based on its goal, i.e. comfort, economy, and safety. Comfort represents typical driver’s assistance systems such as intelligent speed adaptation, traffic sign recognition, wrong-way driving warning, etc. Economy has the goal toward more efficient usage of road space and fuel consumption. For instance, CACC, platooning, highway merge function, etc. Lastly, functions with safety as the goal, usually have the highest complexity due to time and reliability constraints. For example, cooperative intersection collision avoidance systems (CICAS), cooperative lane change, etc.

Cooperative functions may sometimes fulfil two or more goals. The scope of these cooperative goals would have a significant impact on complexity. The scope is divided into three different levels: global, local, and individual. The global scope would give more priority to achieve better traffic flow and reduce congestion. For example, optimizing traffic flow of a whole highway. While, intersection, highway merging, cooperative lane change are examples at the local level. Although some “self-interested” agents could cheat and take advantage of cooperative systems as presented in [27], cooperation to make way for an emergency vehicle is a good example at the individual level.

Cooperative intersection collision avoidance systems (CICAS), cooperative lane change, etc. Economy has the goal toward more efficient usage of road space and fuel consumption. For instance, CACC, platooning, highway merge function, etc. Lastly, functions with safety as the goal, usually have the highest complexity due to time and reliability constraints. For example, cooperative intersection collision avoidance systems (CICAS), cooperative lane change, etc.

Fig. 3: Dimensions of cooperative ITS

Figure 3 illustrate three dimensions of cooperation. The complexity and the need for communication and cooperation grows as the system move away from the origin in any dimension. Automated driving functions help the C-ITS to achieve cooperation and vice versa.

V. CHALLENGES

So far most research within cooperative driving and C-ITS deal with vehicles having the same level of automation. For instance, vehicles in the systems are all equipped with similar sensors, vehicles operate at the same level of automation, in other words, they have the same capabilities. On the contrary, considering real driving situations, cars with different capabilities are mixed in the traffic. There are many challenges already, even in current traffic situations, where driver’s behaviour is a major difference. In the future traffic environment, where new and old cars meet, we would expect vehicles equipped with automated driving applications and communication facilities driving smoothly alongside older vehicles. Besides levels of automation, communicating vehicles from different companies, or different software version of the same cooperative function are other interesting scenarios. Seamless cooperation and interaction between such diverse vehicles are one of the challenges in C-ITS deployment. Apart from cooperation in the systems, interacting with non-cooperative vehicles, telling that there is cooperation, or automation going on, may sometimes be necessary. For example, using a special light signal to inform non-cooperative manually driven vehicles.

Safety is another important issue to address, which can be seen from many perspectives. First, perception failure or malfunction in a vehicle may mislead other vehicles in the system by feeding wrong information into the system. This issue is one example of hazardous events that could occur in the vehicle according to the standard ISO 26262 [28]. With more automation involved in the system as mentioned before, perception failure create risks which could lead to hazardous events. For instance, automated braking systems that suddenly brake the vehicle, or a lane keeping aid function that perform incorrect steering. Another perspective is safety, related to the transition to manual driving if the automation fails. For example, with an automated driving function like CACC, the driver might not always pay attention to the driving tasks. If the system fails and requires the driver to intervene in order to prevent hazardous events, one risk is that the driver is not alert enough to handle the situation. Thus, it could lead to an accident. There are some studies that already considered this issue, for example, [29] propose an architecture that separate applications into manageable and easy to test pieces and also use a communication protocol for collaborative vehicle control.

The larger the C-ITS becomes, the broader and more reliable communication coverage is needed. With such different capabilities as pointed out above and diversities among manufacturers, a standard set of rules are needed, especially in describing the behaviour of cooperative vehicles as elaborated in [24]. The paper used platooning, emergency vehicle warning and intersection scenarios as examples to illustrate lack of common abstraction to describe cooperative vehicle behaviour, for example, it is not clear in the current standard message format how the vehicles should manoeuvre.

One way of representing automation in cooperative driving is to apply the concept of multiagent system (MAS). Agents have suitable characteristics to represent actors in transport systems, which are autonomy, collaboration, and reactivity, as elaborated upon in [30]. Agent technology in traffic and transport systems are presented in [30]. Modelling and simulation are usually the main focus of agent-based applications. However, despite the long list of examples, only a few applications are implemented and deployed in real-world traffic. In conclusion, there are plenty of examples that relate cooperation with automation. Numerous promising simulation results were reported from those projects. Yet,
only few were realized in real-world demonstrators. Thus, closing the gap between the simulation world and the real one is another challenge to be considered.

Last but not least, testing of cooperative driving functions is a challenging task within the area. C-ITS introduce numerous new possibilities and scenarios, testing all of them is nearly impossible. Therefore, ensure that "sufficient" tests have been done is seen as another challenge.

VI. CONCLUSION

This paper first presented C-ITS from two different perspectives: behavioural, and structural. From the behavioural perspective, C-ITS can operate or assist the driver on three levels of driving tasks: a) strategic b) tactical and c) operational. From the structural perspective, components within the actors in C-ITS, in relation to the behavioural perspective, are presented.

Moreover, the main factors to be considered for C-ITS classification are proposed as three dimensions of C-ITS: driving task, scope of goals, and number of actors. The scope of goals is the result of actors’ behaviour and may be limited by the structure of the actors in the system. The number of actors reflects on connections and interaction complexity of the structures in the system. Besides, the relation between the driving task dimension, the behaviour perspective and the planning(time) horizon to solve the driving tasks, is seen as another perspective. Lastly, the challenges in the C-ITS deployment are also elaborated upon.

Interactions between cooperative partners and their drivers are a major contributor to successful cooperation. Especially automated coordination between vehicles with different capabilities is seen as one of the challenges.

In spite of the perspectives and challenges covered in this paper, implementing C-ITS is a great challenge with many issues. The presented analysis of C-ITS and dimensions of driving cooperation in relation to levels of automation is intended to provide a basis for future works regarding C-ITS.

REFERENCES

Appendix B

Paper II
Extended Driving Simulator for Evaluation of Cooperative Intelligent Transport Systems

Maytheewat Aramrattana, Tony Larsson, Jonas Jansson, and Arne Nåbo

2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation
Banff, Alberta, Canada, May 15-18, 2016
Extended Driving Simulator for Evaluation of Cooperative Intelligent Transport Systems

Maytheewat Aramrattana∗, Tony Larsson
Halmstad University
Box 823, SE-301 18 Halmstad, Sweden
{maytheewat.aramrattana, tony.larsson}@hh.se

Jonas Jansson, Arne Nåbo
The Swedish National Road and Transport Research Institute (VTI)
SE-581 95 Linköping, Sweden
{jonas.jansson, arne.nabo}@vti.se

ABSTRACT

Vehicles in cooperative intelligent transport systems (C-ITS) often need to interact with each other in order to achieve their goals, safe and efficient transport services. Since human drivers are still expected to be involved in C-ITS, driving simulators are appropriate tools for evaluation of the C-ITS functions. However, driving simulators often simplify the interactions or influences from the ego vehicle on the traffic. Moreover, they normally do not support vehicle-to-vehicle and vehicle-to-infrastructure (V2X) communication, which is the main enabler for C-ITS. Therefore, to increase the C-ITS evaluation capability, a solution on how to extend a driving simulator with traffic and network simulators to handle cooperative systems is presented as a result of this paper. Evaluation of the result using two use cases is presented. And, the observed limitations and challenges of the solution are reported and discussed.

Keywords

Driving simulator; C-ITS; Traffic simulator; Network simulator

1. INTRODUCTION

Driving simulators have been used to study a variety of topics. For example, the area of human factors, vehicle dynamics, highway design, etc. Driving simulators can be used to evaluate the effect of C-ITS on the human drivers. Since the studies in driving simulators are focused on the human driver, the surrounding traffic is often simplified in a way that the behaviour of the human driver has almost no influence on the traffic. This is not the case in C-ITS, where vehicles are supposed to have interaction with each others. On the other hand, traffic simulators have the capability to model this type of influence between behaviours of vehicles with car-following models. A problem with traffic simulators is that the car-following models usually assume vehicles to follow most of safety criteria and traffic rules. Behaviours such as overtaking on two-lane rural roads or sudden lane changes are often not modelled in traffic simulators. Therefore, combining driving and traffic simulation environments together could produce more realistic ITS traffic models as presented in [11].

Driving and traffic simulators usually do not cover wireless communication, a key enabler to C-ITS, and a big research area by itself as summarized in [17][8]. Open source and widely used network simulators are ns-2 [9], ns-3 [5], and OMNeT++ [15]. However, these do not support modelling of the physical movements of objects, which is the case for vehicles in C-ITS. Therefore, researchers often need to generate trajectories of the vehicles before using them in the simulation. Otherwise, the network simulator has to be coupled with a traffic simulator as done in the Veins [14] framework, which integrates network and traffic simulators.

The three types of simulators mentioned above, i.e. driving, traffic, and network simulators, complement each other and cover most major components of C-ITS, as presented in Fig. 1. The black boxes indicate the components that are not covered in the study in this paper. To the authors'
knowledge, two similar attempts to extend the context of the driving simulator have been presented before. [11] presented the integration of driving and traffic simulation and related issues. Using AIMSUN [1] as the traffic simulation tool and Virtual Environment for Road Safety (VERA) as driving simulator. Aiming for connected vehicle applications design and evaluation, [18] claims that the integrated traffic-driving-network simulator in their work is technically feasible, with a number of challenges to be considered. Their solution used PARAMICS [4] as the traffic simulator, ns-2 as the network simulator, and the University at Buffalo driving simulator.

Background regarding the traffic and network simulator used in this paper is presented in Section 2. An extended driving simulator is presented in Section 3. Section 4 elaborate on use cases of the simulator. Limitations, challenges, and perspectives are discussed in Section 5. Lastly, the paper is concluded in Section 6.

2. BACKGROUND

The traffic and network simulators to be used in the extended simulator presented in the following section is based on PLEXE [13], the platooning extension for Veins. It is chosen for two major reasons. First, it is already coupled with the traffic simulator, Simulation of Urban MOBility (SUMO) [7]. Second, it is made for platooning applications, which in the authors’ opinion, is a C-ITS application that represent a use case with high potential to be deployed in the near future. The author of PLEXE has implemented support for the platooning application in both Veins and SUMO, then released it as plexe-sumo and plexe-veins, denoted as SUMO and Veins in Fig. 2 respectively. The version in plexe-dev branch of Plexe on the Git repository [2] was used in this work. In the original plexe-sumo, there are car-following models implementing: cruise control (CC), adaptive cruise control (ACC), and two cooperative adaptive cruise control (CACC) controllers. The “actuation lag” modelled by a first order low-pass filter is added to the car-following model, to imitate the lag introduced by power-train. In plexe-veins, IEEE802.11p network interface is used by each vehicle. A basic message dissemination scheme and simple platooning application is also provided in plexe-veins.

Therefore, according to Fig. 1 from a driving simulator perspectives, plexe-sumo would cover simulation of surrounding actors and road network. Communication is simulated by plexe-veins. Lastly, cooperative function could be implemented in plexe-veins or driving simulator as elaborated in Section 4.

3. EXTENDED DRIVING SIMULATOR

The extended driving simulator proposed and evaluated in this paper is based on the driving simulator software from the Swedish National Road and Transport Research Institute (VTI) [6], i.e. VTI’s driving simulator, in Fig. 2. This simulation software can run on both desktop environment and more advanced VTI’s driving simulators. Fig. 3 illustrates a screenshot from the driving simulator.

Synchronization

SUMO simulation can be controlled using traffic control interface [16], where SUMO acts as server waiting for client to be connected by setting up transmission control protocol (TCP) connection. When connected, the client, Veins in this case, has to trigger each simulation step in SUMO. Therefore, a similar approach was used to synchronize the driving simulator with Veins and SUMO. Before every simulation time step, the VTI’s driving simulator sends a message to plexe-veins to trigger plexe-sumo to execute one time step as described above.

Modifications

A plug-in, attached to the VTI’s driving simulator software was developed to exchange data with plexe-veins. The plug-in is executed before every simulation time step in the VTI’s simulator. The VTI’s driving simulator is running the plug-in at 100 Hz. Parameters of the vehicles in SUMO, which Veins subscribe to, are forwarded to the VTI’s simulator. That then update position and speed of the vehicles according to the data. Moreover, a control logic can be implemented in the plug-in. The logic calculate speed to control a selected vehicle in SUMO. More details will be elaborate in Section 4.

No change were made on plexe-sumo. A number of changes have been made on plexe-veins to connect with the driving simulator. One TCP connection to the driving simulator was added (referred to as TCPsync in Fig. 2). This connection was used to send vehicles’ position and speed that plexe-veins received from plexe-sumo to the driving simulator. Synchronization between the driving simulator and plexe-veins is done through this connection. Moreover, another TCP connection was added to the application layer in plexe-veins to request and receive control data from the driving simulator (referred to as TCPapp in Fig. 2). Previously all vehicles in the simulation are controlled by car-following model in plexe-sumo. The modification allows driving simulator to control one of the vehicles by sending desired speed through the TCPapp connection. A detailed example is presented in the second use case of Section 4.
4. USE CASES

The proposed extended driving simulator is evaluated with two use cases related to a platooning application. A platoon consists of one leader and one, two, or more followers driving together with a desired gap between the vehicles. Fig. 4 illustrates a platoon of four vehicles together with their parameters, which will be mentioned in this section. The platoon leader is normally represented by index 0. Vehicle $i$ normally refers to the ego vehicle and vehicle $i - 1$ is its preceding vehicle. $x_i$ represents the position of the vehicle $i$. $\text{gap}_{des}$ is the desired gap between the vehicles.

**Use Case 1**

The first use case starts with a platoon of five vehicles running on a straight road. The VTI’s driving simulator visualize the ego vehicle, which is the third vehicle in the platoon, from the driver’s perspective. Two CACC controllers were compared in the simulation: a) CACC, which implements the CACC controller from [12, Chapter 7], and b) Ploeg, the CACC controller presented in [10]. Default communication network parameters in plexe-veins were used. At 40 and 100 seconds, the speed of the platoon’s leader was changed to 25 and 30 m/s respectively. And, after 60 seconds, a command to increase the gap is sent to all vehicles. The two controllers use different concepts of gap between vehicles. CACC implements fixed distance gap with the default value of 10 meters. On the other hand, the Ploeg controller uses the time head way concept, which means the gap is defined by the time that it will take the ego vehicle to reach the position of its preceding vehicle. The default time head way is 0.5 seconds. After receiving the “increase gap” command, the controllers then modify the parameter to 20 meters and 1 second respectively. Fig. 5 and Fig. 6 illustrates and distance between vehicles plotted from data collected in plexe-veins. This use case illustrates the capability of testing different control strategies with human drivers in the loop.

**Use Case 2**

The second use case perform similar scenario, with all vehicles controlled by CACC, except the third vehicle in the platoon. The driving simulator was used to control the third vehicle by executing the control logic shown in (1).

$$
\dot{x}_i = \begin{cases} 
\dot{x}_{i-1} - 5 \text{km/h}, & \text{if } \text{gap}_{des} < 12 \text{meters} \\
120 \text{km/h}, & \text{otherwise}
\end{cases}
$$

(1)

$x_i$ represents desired speed of the ego vehicle and $\dot{x}_{i-1}$ is speed of its preceding vehicle. The desired speed was sent from the driving simulator to plexe-sumo via plexe-veins. The speed then go through the ACC model in plexe-sumo, which also implement the first order low-pass filter for power train modelling as mentioned in Section 2. The result is compared with executing the same control logic in plexe-veins and presented in Fig. 7. The result shows that the simulation environment is synchronized and in phase. The controllers could be executed either by the driving simulator or plexe-veins. This use case elaborate on flexibility of the simulator, where different types of controllers are able to perform together in the same environment, not only limited to car-following models in plexe-sumo.

Figure 4: Platoon of vehicles with parameters

Figure 5: Speed profiles of the vehicles in the “increase gap” scenario.

Figure 6: Distances between vehicles in the “increase gap” scenario.

Figure 7: Plots of distances between vehicles with the ego vehicle of the platoon using the simple control logic in (1).

The horizontal line is drawn at 12 meters.
5. LIMITATIONS AND PERSPECTIVES

One major limitation with the recent version of the proposed simulator is the lack of support for smooth lateral manoeuvres. This is since in traffic simulators like SUMO, the car-following models do not consider lateral acceleration. Consequently, lane changing manoeuvres are modelled as instantaneous, i.e. a vehicle can switch from one lane to another in one time step. Implementing more realistic lateral manoeuvres in the future would allow to perform more complex scenarios, requiring realistic lateral manoeuvres. For example, cooperative lane change, platoon merging, etc. Furthermore, currently the driving simulator does not have access to communication messages in vehicular network. Consequently, C-ITS functions that require data from the network are limited to be implemented only in plexe-veins. Development is needed to make the driving simulator aware of the messages sent in the simulated vehicular network. Lastly, involving a human driver in the scenarios is another evaluation goal of the simulator.

6. CONCLUSIONS

An extended driving simulator for C-ITS evaluation is presented. The VTI’s driving simulator was extended with the plexe-sumo as the traffic simulator, and plexe-veins as network simulator. The thus extended simulator was then evaluated with the use case from a platooning application, implementing an extra ACC controller to elaborate flexibility of the extended simulator. Comparing two CACC controllers from human drivers perspectives is another use case used to illustrate benefits of the simulator. For instance, it may be clear from control theory point of view to only analyse the controllers numerically and with help of plots. But it is hard to judge the human drivers’ preference. With the proposed simulation framework, users acceptance issues can also be studied. Moreover, the limitations of the simulator are presented as issues to be tackled in the future. Although realistic lateral manoeuvres are not available, the simulation framework will still be able to evaluate tactical decisions in C-ITS functions, e.g. decisions when to change lane, but not on the operational level, e.g. the manoeuvres. This work is still ongoing with the goal to answer the following research questions: a) how feasible is this approach with respect to evaluation of C-ITS? and b) what are the other limitations and challenges of this approach?

7. ACKNOWLEDGMENTS

The authors would like to thank Michele Segata from the Universities of Innsbruck (AT) and Trento (IT) for his valuable advices regarding modification and details of Plexe. Also, Jonas Andersson Hultgren from VTI for guidelines on the VTI’s driving simulator software.

8. REFERENCES

Appendix C

Paper III
Cooperative Driving Simulation

Maytheewat Aramrattana, Tony Larsson, Jonas Jansson, and Arne Nåbo

The Driving Simulation Conference 2016 VR
Paris, France, September 7-9, 2016
Cooperative Driving Simulation

Maytheewat Aramrattana 1,2, Tony Larsson 1, Jonas Jansson 2 and Arne Nabo 2

(1) Halmstad University, Box 823, SE-301 18 Halmstad, Sweden
(2) The Swedish National Road and Transport Research Institute (VTI), SE-581 95, Linköping, Sweden

Abstract - For a few decades, driving simulators have been supporting research and development of advanced driver assistance systems (ADAS). In the near future, connected vehicles are expected to be deployed. Driving simulators will need to support evaluation of cooperative driving applications within cooperative intelligent transportation systems (C-ITS) scenarios. C-ITS utilize vehicle-to-vehicle and vehicle-to-infrastructure (V2X) communication. Simulation of the inter vehicle communication is often not supported in driving simulators. On the other hand, previous efforts have been made to connect network simulators and traffic simulators, to perform C-ITS simulations. Nevertheless, interactions between actors in the system is an essential aspect of C-ITS. Driving simulators can provide the opportunity to study interactions and reactions of human drivers to the system. This paper present simulation of a C-ITS scenario using a combination of driving, network, and traffic simulators. The architecture of the solution and important challenges of the integration are presented. A scenario from Grand Cooperative Driving Challenge (GCDC) 2016 is implemented in the simulator as an example use case. Lastly, potential usages and future developments are discussed.

Keywords: C-ITS; Driving simulator; Traffic simulators; Network simulator; Platooning

Introduction

The development of wireless communication is rapidly increasing the means to have vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. Combined with advancement in positioning, vehicles will have improved situational awareness, which will support enhanced functionality of future ADAS and automated driving functions. On the transport system level, this development leads to opportunities for systems that support better cooperation between actors. Such systems is referred to as cooperative intelligent transport systems (C-ITS). In the early development phase, when real equipment and system installations may not exist, simulation is essential to support research and development of the system, especially when evaluating and testing new concepts. While several simulators have been proposed and used in transportation research, there are only few that are dedicated for C-ITS studies. Even fewer include a driving simulator [Zha14, Son15, Pre14].

With respect to C-ITS evaluation, two major limitations of driving simulators are a) they usually do not support V2X communication; and b) surrounding vehicles in driving simulators are often simplified and take almost no consideration of other vehicles. Nonetheless, human drivers are expected to be involved in C-ITS, and driving simulators are feasible tools for human factors studies. To resolve the aforementioned limitations, a concept of extending a driving simulator with existing network and traffic simulators is proposed. The proposed simulation framework is intended for evaluation of C-ITS functions. Details regarding simulators used in this work will be provided in the Background section.

Figure 1: Platoons of vehicles with parameters

Platooning is an example of a C-ITS application that is expected to be implemented soon in real traffic. Sometimes referred to as cooperative adaptive cruise control (CACC), platooning involves two or more vehicles driving in a platoon as shown in Fig. 1. The platoon consists of one lead vehicle (leader) in front, and one or more vehicle following the leader, while keeping a desired safe distance \(\text{gap}_{\text{des}}\) between vehicles. Vehicles driving close to each other in a platoon is expected to reduce fuel consumption and provide better utilization of road spaces, as well as driver's comfort.

In this paper, a simplified version of a platoon merging scenario from the Grand Cooperative Driving Challenge (GCDC) 2016 [GCD], where two platoons have to communicate and merge safely into one platoon, is chosen as a test scenario. The scenario elaborate on benefits of having V2V communication, in additional to driving automation. Details regarding the scenario is described in the Scenario section. Moreover, methods used to extend the driving simulator are presented in the Integration section. The results section presents two evaluation examples using the extended driving simulator. Usage, challenges,
and future plans are discussed in the Discussion section.

Background
This work integrates an existing driving simulator from the Swedish National Road and Transport Research Institute (VTI) with Plexe [Seg14]; a traffic and network simulator mainly aimed for simulation studies of platooning scenarios. The traffic simulator used in Plexe is based on Simulation of Urban MOBility (SUMO) [Kra12]. And, the network simulator is based on Veins [Som11].

Figure 2 illustrates a decomposition into simulation software packages used for the C-ITS simulator. The components in black boxes are desired but not modelled in this paper.

VTI’s Driving Simulator
The VTI simulator is made up of mainly C++ components and has been developed in house at VTI. It is implemented for distributed simulation and can be divided into three main modules: VISIR - the graphics rendering, SIREN - the sound software and CORE - the kernel software running the main simulation loop. CORE also include the vehicle dynamics model, scenario description, cabin interface and HMI software. The software can be executed either in desktop environment or in simulators with motion system at VTI such as Sim IV [Jan14]. In this paper, the software only run in a desktop computer.

Plexe
Plexe consists of SUMO and Veins. SUMO is an open source microscopic traffic simulation software. A number of models has been added to SUMO (version 0.22) in its Plexe version (plexe-sumo). Models of cruise control (CC), adaptive cruise control (ACC), and cooperative adaptive cruise control (CACC) using car-following models are available. Two CACC controllers are available in Plexe: a) the CACC controller from Rajamani’s book [Raj12]Chapter 7, refer to as CACC in this paper; and b) the CACC controller proposed by Ploeg et al. [Plo11], refer to as Ploeg in this paper. Microscopic traffic models implement a driving behaviour for each vehicle. This driving behaviour contains variations and some randomness, i.e. non-deterministic behaviour. However, the models used in this paper are intended to simulate vehicles in automated driving mode. Thus, behaviour of the vehicles are deterministic.

To model the behaviour of the power-trains in vehicles, a first order low-pass filter is used as shown in Fig. 3.

![Figure 3: A low-pass filter applied to the vehicles in simulation.](image)

Veins is a vehicular network simulation framework, built on top of an event-based network simulator, OMNeT++. Since network simulators typically do not consider mobility of communication nodes, Veins interacts with SUMO to obtain movements of the nodes. It can also be used to control vehicles in SUMO. This interaction is done using Traffic Control Interface (TraCI) [Weg08]. Moreover, the Plexe version of Veins (plexxe-veins) provides IEEE 802.11p network interface to each vehicle. Also, a simple platooning application using a basic message dissemination scheme is provided in plexxe-veins.

C-ITS Simulation Framework
The connections between VTI’s driving simulation software and Plexe is shown in Fig. 4. Two transmission control protocol (TCP) connections are established between the driving simulation and Plexe. Besides exchanging information between SUMO and Veins, synchronization between them is done using TraCI. An existing server-client TCP connection, TraCI in Fig. 4, is used between SUMO and Veins, as server and client respectively. At each time step, Veins trigger SUMO to execute one time step with a TraCI message. SUMO then return vehicles’ information that Veins subscribed e.g., speed, position, etc. Therefore, a TCP connection, TCP_sync, is added for synchronization between the VTI’s driving simulation software and Plexe in a similar way. The driving simulation trigger the process between Veins and SUMO. Afterwards, the received vehicles’ information in Veins are forwarded to the driving simulator. All simulators are running in real-time with 0.01 second time step.

Another TCP connection, TCP_app, is created for exchanging information from the application layer of plexxe-veins to the VTI’s software. For example, exchange information with a control logic implemented in the driving simulator to control a vehicle in the simulation.

In the VTI’s driving simulation software, a plug-in was
developed. It receives information such as vehicle name, speed, and position (Cartesian coordinate system). One of the vehicles in the simulation is chosen as ego vehicle. The driving simulation software then visualize perspectives from the ego vehicle. Moreover, it implements realistic visualization of lane change manoeuvres. Since the car-following models in SUMO does not consider lateral acceleration, lane changes in SUMO occur instantaneously. Vehicles switch from one lane to another in one time step. When a lane change occur in SUMO, the plug-in in VTI’s driving simulator output a smooth lane changing manoeuvre. The manoeuvre is executed using a PID controller that output yaw velocities based on error in lateral position.

Furthermore, to change lane in SUMO, a lane changing model is implemented in each vehicle. When the vehicle is told to change lane, the model make a decision whether to change lane or not. It considers the space between vehicles in the other lane. If the space is not large enough, the vehicle will not change lane and try to speed up and overtake. Since vehicles are driving fairly close to each other in platooning applications, the lane changing model in SUMO often results in vehicles trying to speed up and overtake the platoon leader. Therefore, the lane changing model in SUMO is modified, to make decisions without considering the space in other lane. Hence, vehicles change lane immediately, when they are told to do so.

Platoon Merging Scenario

As an example use case of the simulation framework, a simplified version of platoon merging scenario from the GCDC competition is implemented. There are two platoons in the scenario, platoon A and platoon B. The implementation is described as a state machine diagram in Fig. 5 for platoon A and Fig. 7 for platoon B.

Platooning

The scenario starts with two platoons driving at 60 km/h on a two-lane highway. “platoon A” driving on the left lane, “platoon B” driving on the right lane. The leader of each platoon is set as the first vehicle (FV). Each platoon is led by the organizer’s pace car (OPC), which will not participate in the merging and is not considered as a part of platoon. Desired distance between vehicles is defined in Eq. 1.

\[ CACC_{gap} = r + hv \]  

\( CACC_{gap} \) is a desired inter-vehicle distance in meter, \( r \) is a standstill distance (6 meters is used as suggested in the GCDC documentation [Sal16]), \( h \) is time headway, and \( v \) is velocity of the vehicle in m/s.

B2A Pair-up

Next, the two platoons receive information about a road-work ahead on the left lane. Consequently, \textit{platoon A} request to merge with \textit{platoon B}, initiate two pair-up phases illustrated in Fig. 6. Starting from vehicles in the \textit{platoon B}, each vehicle find its forward-pair (FWDPair) simultaneously. Forward-pair vehicle is a vehicle in the other platoon, that is ahead of ego vehicle and behind the preceding vehicle of the ego vehicle. Vice versa for a backward-pair (BWDPair). A vehicle is supposed to keep a desired distance to its forward-pair in order to make a gap for the forward-pair vehicle to merge in front of it.

A2B Pair-up

When the first vehicle in \textit{platoon A} has realized that another vehicle has selected it as its forward-pair, it selects that vehicle as its backward-pair. Afterwards, it start to search for its own forward-pair. After a forward pair is found, the vehicle and its backward pair start making a gap. When the gap is ready, the vehicle in \textit{platoon A} set its mergingFlag and enter “wait STOM” state. Then, it passes the firstVehicleFlag back to the next one in the platoon. The A2B
pair-up and make gap processes are then repeated until everyone makes a gap to its forward pair, sequentially.

Make Gap

Once a vehicle in platoon A has both forward and backward pairs, the vehicle and its backward pair will start to make a gap. Equation 2 is used for making the gap.

\[ \text{gap}_{\text{desired}} = \text{gap}_{\text{current}} + K_p \times (\text{gap}_{\text{safe}} - \text{gap}_{\text{FW}}) \]  

\text{gap}_{\text{desired}} \text{ is the desired } C\text{ACC}g_{\text{ap}}, \text{gap}_{\text{current}} \text{ is the current distance to the vehicle in front, } K_p \text{ is the controller gain, } \text{gap}_{\text{safe}} \text{ is the desired distance to the forward pair, and } \text{gap}_{\text{FW}} \text{ is the current gap between ego vehicle and its forward pair. Variables are illustrated in Fig. 6.}

Merge

Eventually, the platoons enter the merging zone at 2000 meters, where vehicles from platoon A change lane to merge with platoon B. However, before a vehicle can merge, it has to assure that the safe-to-merge (STOM) flag in its backward-pair’s iCLCM is set.

Platoon reformation

After a vehicle in platoon A has merged, the platoon is reformed. All gaps are reduced to the normal desired distance used in the platooning phase. After the merge, the vehicle is now following its forward pair. Therefore, MIO is now changed to the forward pair. Since in this paper we assume that all vehicles start merging as soon as they receive STOM, a vehicle in platoon B immediately change its MIO to its forward pair after it transmitted STOM. Thus, following the vehicle that is merging in front of itself. For a vehicle platoon A, once STOM is received, it will change lane and its MIO to the forward pair. Vehicles in both platoons then reset their CACCgap to the starting value.
simulation run for 200 seconds, starting with two platoons; five vehicles on the right lane; four vehicles on the left lane. Figure 9 illustrates the two platoons with vehicles’ identification. The two platoons are driving at 60 km/h and travel for about 3300 meters. At the simulation time 50 seconds, the merge request message is sent out and the platoons start the pair-up process. The first vehicle enter the merging zone at 2000 meters (about 150 seconds in simulation time). The platoons merged and reformed to one platoon, and drive until the end of the simulation. Overview of the simulation is illustrated in Fig. 8.

A scenario with $CACC_{gap}$ in Eq. 1 equals to 11 meters is simulated. The desired gap in Eq. 2 of 20 meters is selected. In the simulation, after it has found the pairs, each vehicle in platoon $A$ has 25 seconds to make the gap. After 25 seconds, the gap is assumed to be done. The vehicle then travel at a constant speed of 60 km/h, stop tracking the distance to its forward pair, and pass the firstVehicleFlag backwards to the next vehicle.

Figure 10 presents a plot of distance to the object in front of each vehicle. The plot is the simulation result of $CACC$ controller when $11$ meters $CACC_{gap}$ and $K_p = 2.0$ are used. Since vehicle number 0 and 1 are OPCs, and vehicle 2 did not make gap, they are omitted from the plot.

Evaluation of gap-making

Moreover, gap from each vehicle to its forward pair during gap-making phase, with $K_p = 1$, 2, and 3, in Eq. 2 is presented in Fig. 11, 12, and 13, respectively. From the plots, when $K_p = 1$, vehicles are not able to make desired gap within the given time (25 second). While $K_p = 2$ and 3 fulfil the requirement, $K_p = 3$ caused higher variation in speed compared
Figure 11: Distance between the vehicle and its forward pair \( \text{gap}_{FW} \) in Fig. 6. \( \text{CACC} \text{gap} = 11, K_p = 1 \)

Figure 12: Distance between the vehicle and its forward pair \( \text{gap}_{FW} \) in Fig. 6. \( \text{CACC} \text{gap} = 11, K_p = 2 \)

Figure 13: Distance between the vehicle and its forward pair \( \text{gap}_{FW} \) in Fig. 6. \( \text{CACC} \text{gap} = 11, K_p = 3 \)

to \( K_p = 2 \) as shown in Fig. 14. In all cases, there is not enough time for vehicles in platoon \( A \) to stabilize the gap. Nevertheless, it shows that the simulation framework can be used to support design of gap-making strategy. Moreover, human factors aspects of the strategy can be studied with the driving simulator.

Evaluation of CACC controllers

Another simulation with the vehicles controlled by \( \text{Ploeg} \)’s CACC controller, which use time headway policy, is simulated. The desired gap between vehicles \( \text{CACCgap} \) are fixed to 11 meters and \( K_p = 1 \). However, the standstill distance, \( r \) in Eq. 1, of 2 meters is implemented for \( \text{Ploeg} \). Therefore, the time headway during platooning phase is 0.54 second. And, \( \frac{\text{speed}_{planned}}{\text{speed}} \) is used as the time headway while making gap, where \( \text{speed}_{desired} \) is the result of Eq. 2 and \( \text{speed} \) is current velocity of ego vehicle in m/s. Figure. 15 illustrates plots of distance to the object in front of each vehicle for both controllers.

From the first 50 seconds of Fig. 15, \( \text{Ploeg} \) closes the gaps between vehicles and go down to the desired distance, 11 meters, faster than \( \text{CACC} \). On the other hand, during the gap-making phase, \( \text{CACC} \) shows smoother gap-making manoeuvre compare to \( \text{Ploeg} \). Hence, we can conclude that the gap-making strategy presented in Eq. 2 is less suitable with \( \text{Ploeg} \)'s controller. It causes big variations in speed, result in vehicles need to accelerate and de-accelerate more than necessary.

Discussion

Despite potential benefits presented in the previous sections, the proposed C-ITS simulation framework is still under development. Limitations and future plans will be discussed in this section. First, the VTI’s driving simulation software is currently used as a visualization tool and not aware of V2V messages in pulse-veins. Once this feature is implemented, scenarios involving interaction with human drivers will be possible to run. Moreover, all vehicles displayed in the driving simulation software are controlled by SUMO. Providing the opportunity to have a human driver driving manually in the scenario will be a valuable addition to the framework, and enable more scenarios including human drivers.

Smooth lane changing manoeuvre is solved only vi-
ualy in the VTI’s driving simulation software. Evaluation of controllers used to solve tactical driving tasks, e.g. decision on when to change lane, is possible. However, analysis on driving tasks at the operation level, e.g. how the controller execute lane changing manoeuvre, is not possible at the moment. This limitation is due to SUMO does not consider lateral acceleration of vehicles. Therefore, lane changes occur instantaneously. Implementing a PID controller, similar to the one used in the driving simulation software for lane changing, is one alternative to solve this limitation. Alternatively, vehicles can be controlled by the driving simulator or other external sources during the lane change. Lastly, selecting another traffic simulator that support lateral acceleration is also an option.

Last but not least, the simulation framework needs to be validated. The GCDC 2016 requires all competing vehicles to log their vehicle’s information such as speed, global positioning system (GPS) coordinates, etc. Therefore, using the logged data to validate the simulation framework is an essential future work. The implementation of the scenario in this paper is mostly based on the data provided by the GCDC organizer before the competition. A few modifications have been made from the document in the real competition. Therefore, in order to validate the simulation with the logged data, more details need to be implemented.

Conclusion

To support evaluation of C-ITS functions, driving simulators need to support V2X communication, and the interactions between ego vehicle and its surroundings. This paper presents a C-ITS simulation framework consisting of 3 separate software packages integrated into a C-ITS driving simulator: a) the driving simulation software from VTI; b) Plexe version of SUMO, the microscopic traffic simulator; and c) Plexe version of Veins, the inter-vehicular network simulation framework.

There are still several open questions concerning the architecture and how to integrate the components, e.g. in this work the the driving simulator software ensures realistic/physically correct behaviour of each vehicle. Another alternative would be to include more realistic models in the traffic simulation software.

The platoon merging scenario from the GCDC 2016 was selected as a use case, because of the opportunity to validate the simulation framework with real-world data collected during the competition. Moreover, the scenario has a good mixture of cooperative and automated driving.

During transition to fully automated and cooperative driving, human drivers will still be involved in the driving task, either by assisted-driving or monitoring the vehicles. Therefore, involving human in the loop is necessary for the design of future ADAS. Although this paper did not perform any study including a human driver, it presents one way of involving the driver in C-ITS simulation. The presented results elaborated on a few aspects of evaluating a C-ITS function. The simulation framework can also be used to study other aspects of C-ITS. For instance, effects from communication disturbance on C-ITS functions. With further development to enable manual driving and interaction from human driver, the simulator would be feasible for studies on human factors in C-ITS.

The proposed framework is not limited to the VTI’s driving simulation software. A driving simulator software that support external connection would be able to integrate with Plexe in a similar way. Apart from the driving simulation software, the rest of the framework is open-source. Hence, further extension can be made to incorporate models from other tools and programming environments.

Acknowledgement

The authors would like to thank Michele Segata from the Universities of Innsbruck (AT) and Trento (IT) for his valuable advices regarding modification and details of Plexe. Also, Jonas Andersson Hultgren and Bruno Augusto from VTI, for their guidances on the VTI’s driving simulation software. This work is supported by SAFER – Vehicle and Traffic Centre at Chalmers, as a part of Vehicle ICT Innovation Methodology (VICTiM) project.

References


