Michele Segata

SAFE AND EFFICIENT COMMUNICATION PROTOCOLS FOR PLATOONING CONTROL

PhD Thesis

Advisors

Prof. Dr.-Ing. Falko Dressler

Prof. Renato Lo Cigno

Institut für Informatik
University of Innsbruck

Department of Information Engineering
and Computer Science
University of Trento

UNIVERSITY OF TRENTO - Italy

universität
innsbruck

Michele Segata

Safe and Efficient Communication Protocols for Platooning Control

PhD Thesis

February 2016

UNIVERSITY OF TRENTO - Italy

University of Trento
Department of Information Engineering and Computer Science
Via Sommarive 9, I-38123, Povo (TN)

universität
innsbruck

University of Innsbruck
Department of Computer Science
Technikerstrasse 21a, A-6020, Innsbruck

PhD Advisors	Prof. Dr.-Ing. Falko Dressler Prof. Renato Lo Cigno
Defense committee	Univ.-Prof. Justus Piater (chairman) Prof. Dr.-Ing. Falko Dressler Dr. Francesco Gringoli Prof. Renato Lo Cigno (secretary)
Defense date	Friday, February 12, 2016 – 2.00 pm
Location	ICT building, Technikerstrasse 21a, A-6020, Innsbruck, Room 3W04

ISBN (printed)	978-88-8443-667-2
ISBN (digital)	978-88-8443-668-9
Editor	Università degli Studi di Trento
Published	February 2016
Cover design	© Michele Segata
Printed by	Tipografia Unitn

To Caroline and to the new member of our family.

"Science is a way of life. Science is a perspective. Science is the process that takes us from confusion to understanding in a manner that's precise, predictive and reliable - a transformation, for those lucky enough to experience it, that is empowering and emotional."

— Brian Greene

Abstract

Modern vehicles are becoming smarter and smarter thanks to the continuous development of new Advanced Driving Assistance Systems (ADAS). For example, some new commercial vehicles can detect pedestrians on the road and automatically come to a stop avoiding a collision. Some others can obtain information about traffic congestion through the cellular network and suggest the driver another route to save time. Nevertheless, drivers (and our society as well) are always striving for a safer, cleaner, and more efficient way of traveling and standard, non-cooperative ADAS might not be sufficient. For this reason the research community started to design a vehicular application called "*platooning*". Platooning simultaneously tackles safety and traffic congestion problems by cooperatively coordinating vehicles in an autonomous way. Traffic flow is optimized by using an advanced Adaptive Cruise Control (ACC), called Cooperative Adaptive Cruise Control (CACC), which drastically reduces inter-vehicle gaps. By being autonomously coordinated, platooning vehicles implicitly implement automated emergency braking, a fundamental application for freeway safety.

The idea is to form optimized road trains of vehicles where the first drives the train, while the others autonomously follow at a close distance, without requiring the driver to steer, accelerate, or brake. Platooning can have an enormous impact on future transportation systems by increasing traffic flow (and thus reducing congestion), increasing safety, reducing CO_2 emissions, and reducing the stress of driving. This application is extremely challenging due to its inter-disciplinary nature. Indeed, it involves control theory, vehicle dynamics, communication, and traffic engineering. In this thesis we are mostly concerned with the communication aspects of this application, which is fundamental for making the vehicles cooperate, improving the efficiency of the application with respect to a pure sensor-based solution. Application requirements are very tight and, given that the envisioned communication technology will be IEEE 802.11-based, there are concerns on whether these requirements can really be met. The focus of the thesis is in this direction.

The first contribution is the design of PLEXE, an extension for the widely used vehicular simulation framework Veins that enables research studies on various platooning aspects, including design and evaluation of control algorithms, communication protocols, and applications. The tool is open source and free to download and use, and it realistically simulates both communication and vehicle dynamics. This makes PLEXE a valid testing platform before real world deployment.

The second contribution is a set of undirected information broadcasting (beaconing) protocols that specifically take into account the requirements of the application. We initially develop four static (i.e., periodic) approaches and compare them against two state of the art dynamic protocols, showing that our approaches are capable of supporting the application even in heavily dense scenarios. Then, we propose a dynamic protocol that further improves the application (increasing safety) and the network layer (reducing resource usage) performance.

The final contribution is a platooning control algorithm that, compared to state of the art approaches, is re-configurable at run-time and that can be adapted to network conditions. We thoroughly test the algorithm in highly challenging scenarios. These scenarios include a realistic network setup where the road is shared by human- and automated-driven vehicles. Human-driven vehicles interfere with automated-driven ones by sending data packets on the same channel. Moreover, we also consider a scenario with realistic vehicle dynamics, which takes into account vehicles' engine and braking characteristics. The algorithm is shown to be robust to network and external disturbances, to have a fast convergence, and to be very stable.

The results in this work thus represent a big step towards the real world implementation of platooning systems.

Contents

Abstract iii

Nomenclature vii

1 Introduction 1

2 Fundamentals and Background 11

2.1 Vehicular Communication . . . 11

2.1.1 Communication Technologies, Standards, and Applications . . . 11

2.1.2 Congestion Control in Vehicular Networks . . . 15

2.2 Control Systems . . . 23

2.2.1 Actuation Lag . . . 23

2.2.2 String Stability . . . 25

2.2.3 Cruise Control . . . 25

2.2.4 Adaptive Cruise Control . . . 26

2.2.5 Cooperative Adaptive Cruise Control . . . 28

2.3 Realistic Vehicle Modeling . . . 31

2.3.1 Engine Acceleration and Lag . . . 32

2.3.2 Brakes Deceleration and Lag . . . 35

2.4 Review of Existing Platooning Simulation Tools . . . 35

3 Simulation Tool: PLEXE 39

3.1 Implementing Platooning Capabilities in SUMO . . . 42

3.2 Platooning Protocols and Applications in Veins . . . 45

3.3 Platoon Maneuvering . . . 46

3.4 Sample Use Cases . . . 47

3.4.1 Controller Analysis . . . 48

3.4.2 Join Maneuver 49
3.4.3 Human-driven Vehicles Interference 50
3.4.4 Engine Model 54
3.5 Conclusion 55

4 Safe and Efficient Communication for Platooning 57
4.1 Application Layer Requirements 58
4.2 Experimental Validation 60
4.3 Analysis of STB and SLB 62
4.3.1 Generic Network Performance 64
4.3.2 Application Layer Perspective 66
4.3.3 Impact of CCA-threshold 68
4.3.4 Coexistence with DCC 70
4.3.5 Impact of Communication on CACC Performance 70
4.4 A New Dynamic Approach: Jerk Beaconing 72
4.4.1 Lower Layer Reliability 74
4.4.2 Performance Analysis 77
4.5 Conclusion 82

5 A Consensus-based Controller 85
5.1 Background and Motivation 86
5.2 Consensus-based Control 87
5.3 Evaluation 90
5.3.1 Network and Vehicular Traffic Scenario 90
5.3.2 Basic Convergence Analysis 93
5.4 Analysis with Realistic Vehicle Dynamics 94
5.5 Conclusion 98

6 Conclusion and Outlook 101

Bibliography 115

Nomenclature

Symbol	**Unit**	**Description**
a_{ij}	–	Control topology matrix (vehicle j to i) (Consensus' CACC)
A_L	m^2	Vehicles' cross section
b_i	–	Speed error gains (Consensus' CACC)
C_1	–	Leader/front vehicle weighting factor (PATH's CACC)
c_{air}	–	Vehicles' drag coefficient
cr_1	–	Rolling resistance intercept parameter
cr_2	–	Rolling resistance speed parameter
d_d	m	Desired distance (PATH's CACC)
d_{wheel}	m	Tractive wheels diameter
η	–	Engine and driveline efficiency
F_A	N	Air friction
F_{brake}	N	Brakes force
F_{eng}	N	Engine force at wheel
F_F	N	Friction forces acting on the vehicle
F_G	N	Gravitational force
F_R	N	Rolling resistance
γ	–	Wheel drive coefficient

g	m/s^2	Gravitational acceleration
h_{ij}	s	Time headway (vehicle j to i) (Consensus' CACC)
i_d	–	Differential (final) ratio
i_g	–	Gear ratio
k_d	–	Derivative gain (Ploeg's CACC)
k_i	–	Integral gain (CC)
k_{ij}	–	Distance error gains (Consensus' CACC)
k_p	–	Proportional gain (CC and Ploeg's CACC)
λ	–	ACC design parameter
λ_m	–	Mass factor for driveline rotational inertia
l_i	m	Length of the i-th vehicle in the platoon
μ	–	Tires friction coefficient
m	kg	Mass of the vehicle
N_C	–	Number of engine cylinders
N_{eng}	rpm	Engine speed
n_{eng}	rps	Engine speed
ω_n	Hz	Bandwidth (PATH's CACC)
P_{eng}	W	Engine power
ρ_a	kg/m^3	Air density
r_{wheel}	m	Tractive wheels radius
T	s	Time headway (ACC and Ploeg's CACC)
T_{eng}	N m	Engine torque
τ	s	Actuation lag (first order lag)
τ_{brake}	s	Brakes actuation lag

τ_{eng}	s	Engine actuation lag
θ_{road}	°	Road slope
u_i	$\mathrm{m/s^2}$	Control input (i.e., desired acceleration) for the i-th vehicle in the platoon
x_i	m	Position of i-th vehicle in the platoon
$\dot{x}_{\mathrm{d}}$	m/s	Desired speed (CC)
$\dot{x}_i$	m/s	Speed of the i-th vehicle in the platoon
$\ddot{x}_i$	$\mathrm{m/s^2}$	Acceleration of the i-th vehicle in the platoon
ξ	–	Damping ratio (PATH's CACC)

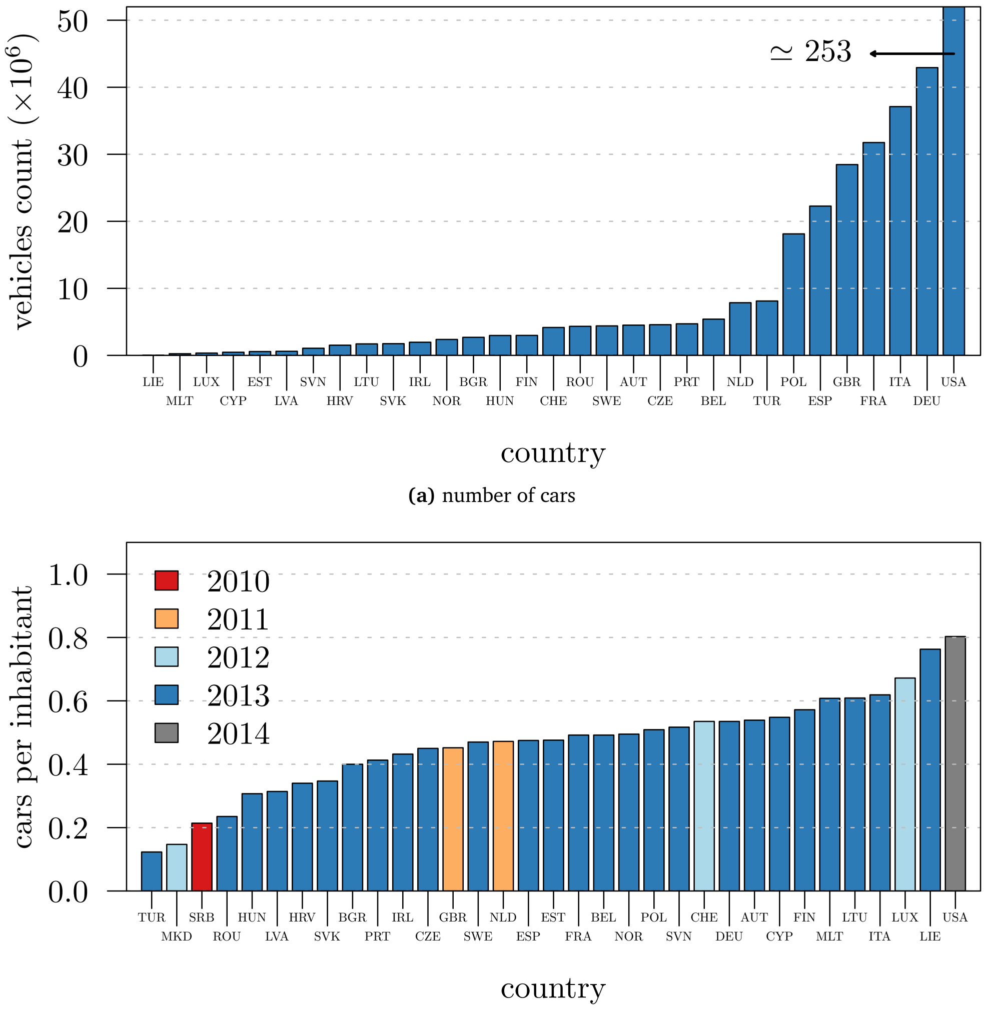
Chapter 1

Introduction

With no doubts we can say that the innovation of vehicles brought several benefits to humanity. Thanks to them people can travel fairly long routes in relatively short times, they can transport goods from one place to another, and can offer services such as public transportation. Cars (and vehicles in general) have become an integral part of the life of people in modern developed countries. This is demonstrated in Figure 1.1, which shows the total number of cars and the number of cars per inhabitant for different European countries and for the United States. In Europe, the three top countries are France, Italy, and Germany, with an amount of vehicles around 30 to 40 million. The United States, being much larger than European countries, counts roughly 253 million vehicles. Figure 1.1b is even more impressive, because it shows that in most of the European countries and in the United States there is at least one vehicle every two inhabitants, showing how common and wide-spread the usage of vehicles has become.

On the other hand we are aware of the downsides of vehicle transportation, i.e., accidents, traffic, and pollution. Even if the number of fatalities per year is continuously decreasing [4, 7], the amount is still not negligible. In 2011, European countries such as Italy, France, Germany, and Poland counted around 4000 deaths due to traffic accidents (Figure 1.2a). In the United States, the amount of fatalities in the same year was around 30 000 people. The statistics show that the majority of the fatalities occur on high speed roads such as motorways and rural streets. In 2000, roughly 68 % of the fatalities in the EU occurred on high speed roads [23], while in 2011, this percentage was around 62 % [7] (Figure 1.2b).

Another important problem is traffic congestion, which affects the efficiency of transportation systems and hence fuel consumption and pollution. According to the statistics

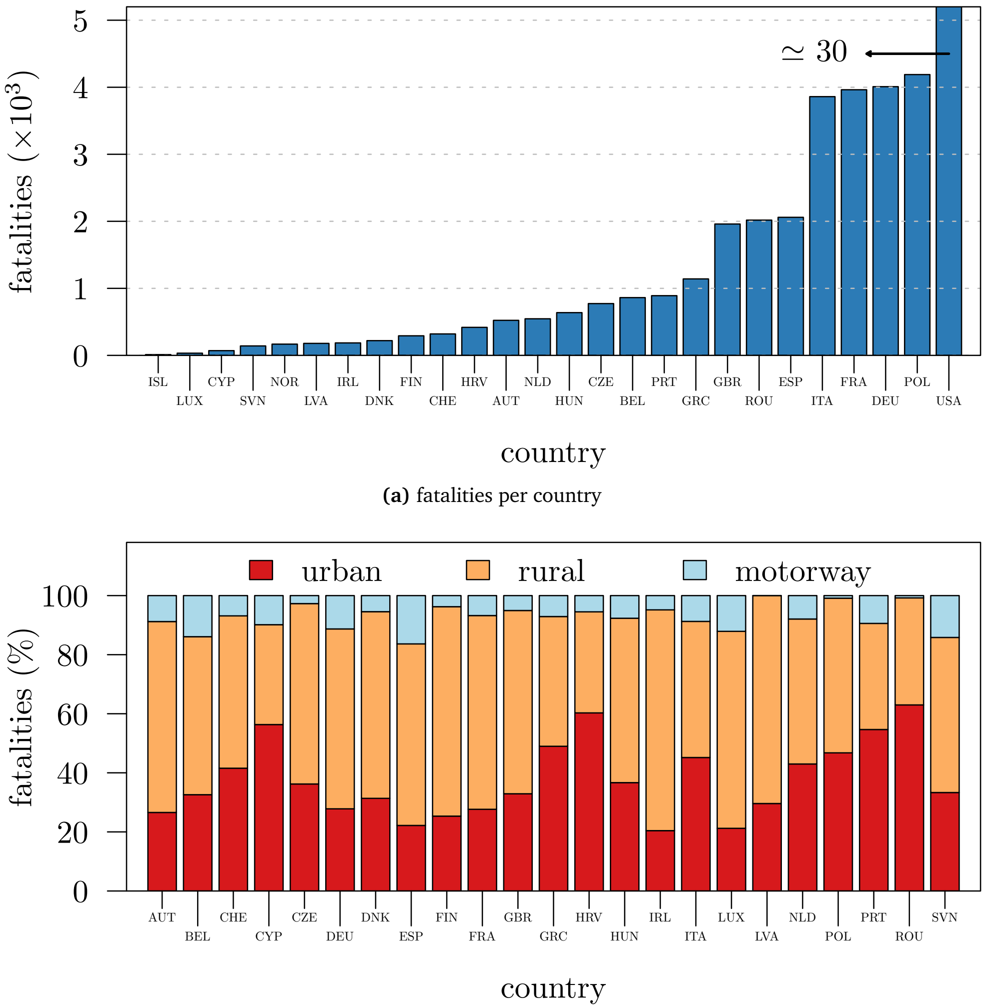


(a) number of cars

(b) cars per inhabitant

Figure 1.1 – Statistics on the total number of cars per country and the number of cars per inhabitant. Sources EUROSTAT [5], U.S. Departments of Transportation [6], and U.S. Census Bureau [22].

published by the EU, about 20 % of the greenhouse gas emissions are due to transportation, 70 % of which are due to road transportation. Moreover, the EU also estimated that congestion costs roughly 1 % of the GDP to Europe every year [2]. Vehicles have thus a major impact on the ecosystem of the Earth and on the economy; Thus by improving the efficiency of the road transportation system we can reduce the amount of CO_2 we produce and improve quality of life.

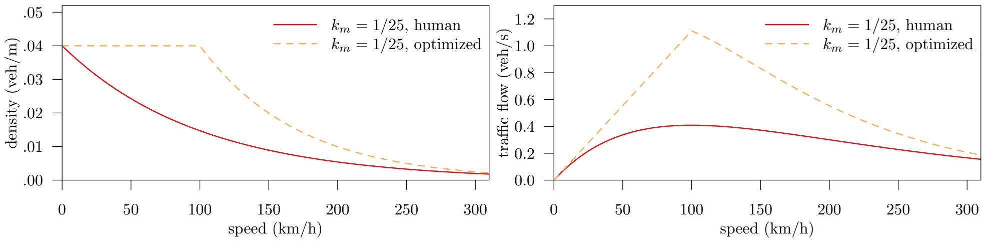


(a) fatalities per country

(b) fatalities per country and road type

Figure 1.2 – Fatalities count and fatalities split by road type for year 2011. Sources European Commission [7] and NHTSA [4].

Typical phenomena caused by congestion are “traffic shockwaves”. These phenomena take place due to zones of transition where the flow changes from high-speed flowing to congested (steady) flowing and vice versa. Such transition zones propagate backward with respect to the motion of vehicles forming, indeed, a wave. Besides being dangerous for drivers, shockwaves cause unnecessary deceleration and acceleration, and thus a waste of fuel and a consequent increase of greenhouse emissions [118]. To reduce traffic



(a) density **(b)** flow

Figure 1.3 – Traffic flow improvement by increasing maximum density for a free-flow speed $v_m = 100\,\mathrm{km/h}$.

shockwaves we need to reduce road congestion, which occurs when the traffic demand is larger than the capacity of the road [30], i.e., if the number of vehicles per second that enter a stretch of road is larger than what the road can handle. To compute the capacity of a road (per lane) we need to consider the traffic flow equation [30], i.e.,

$$q = k \cdot v, \tag{1.1}$$

where q is the flow, and k and v the density (in vehicles per m) and the speed (in m/s) of the vehicles, respectively. To compute the density we need to count the amount of vehicles per unit of space. Assuming all vehicles have the same length and maintain the same distance, the maximum density per lane is computed as

$$k = \frac{1}{l + T \cdot v} \quad , \tag{1.2}$$

where l is the average vehicle length and the product of headway time T and speed v represents the distance. The inter-vehicle gap, according to legislation, must increase with speed for safety reasons. Each driver should maintain a headway time T in the order of 2 s, which might be increased depending on driving conditions. If we substitute Equation (1.2) in Equation (1.1) we obtain

$$q = \frac{v}{l + T \cdot v}. \tag{1.3}$$

To increase the flow we can try increasing the speed. Equation (1.3), however, has a horizontal asymptote, i.e.:

$$\lim_{v \to +\infty} \frac{v}{l + T \cdot v} = \frac{1}{T} \quad , \tag{1.4}$$

which means that $\frac{1}{T}$ is the upper-bound for the flow when using a headway time spacing policy. If the demand is higher than $\frac{1}{T}$ vehicles per second, we have congestion. Furthermore, real traffic flows do not behave as in the ideal model of Equation (1.3). Usually the traffic flow increases up to a maximum value around free-flow speed. For speeds above the free-flow speed the flow starts to decrease because vehicles are getting sparser and sparser. In particular, Greenberg [40] modeled the density as a negative exponential function with respect to cruising speed:

$$k(v) = k_m e^{-\frac{v}{v_m}}. \tag{1.5}$$

In Equation (1.5), k_m is the density at standstill and v_m the free-flow speed. The resulting flow is thus

$$q(v) = v \cdot k(v) = v \cdot k_m e^{-\frac{v}{v_m}}. \tag{1.6}$$

The solid lines in Figures 1.3a and 1.3b show density and flow as a function of speed for the Greenberg model, for a free-flow speed $v_m = 100\,\text{km/h}$.

For both the idealized and the Greenberg model the only possibility to increase the flow and decrease the congestion is to reduce the inter-vehicle spacing, i.e., increasing the part of utilized road. If, by means of an automated system, we would be able to maintain a constant density up to the desired free-flow speed, the gain in maximum flow capacity would be enormous. Consider the optimized vehicle density function in Figure 1.3a, i.e., a constant density k_m up to the free-flow speed v_m. For such a density function, the maximum flow would be more than doubled (Figure 1.3b).

The reason behind this is that the safety distance needed to account for human reaction time causes a huge waste of road infrastructure. To understand the actual fraction of utilized road we can simply compute the ratio between the utilized road length and the overall road length, i.e.:

$$\text{maximum utilization (\%)} = \frac{l}{l + T \cdot v} \cdot 100. \tag{1.7}$$

Figure 1.4 plots the maximum highway utilization as a function of speed for different values of l and T. When respecting safety distances ($T = 2\,\text{s}$) and driving at 100 km/h we are using less than 30 % of the available road. If we only consider cars (4 m long) the maximum utilization reduces to 7 % only. This proves how crucial it is to reduce the gap between consecutive vehicles.

Reducing the inter-vehicle gap has an additional impact on fuel savings because of air drag. Figure 1.5a shows the resistive force caused by the air. The resistive forces are generated by the high-pressure zones in the front of the car and by the turbulent

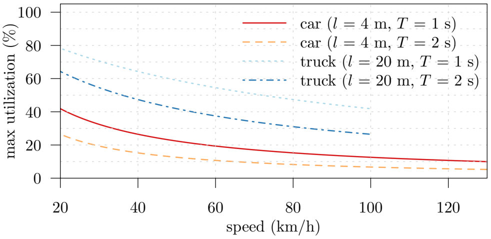
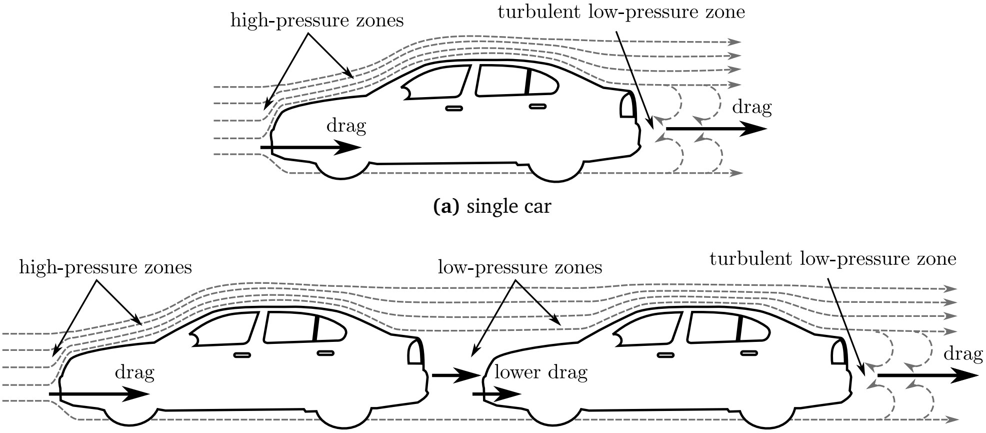


Figure 1.4 – Maximum road utilization as function of speed when respecting safety distance.



(a) single car

(b) two close cars

Figure 1.5 – Air drag for a single car and two cars traveling close to each other.

low-pressure zone, which forms at the tail, "pulling" the car back. When two vehicles travel very close one another (Figure 1.5b) the first car is subject to a lower rear drag because the second car disrupts the turbulent flow. The second car, instead, suffers a lower front drag because it is traveling in a lower density fluid. The larger the group of vehicles, the higher is the overall fuel saving. A study performed in the scope of the European SARTRE project [66] has shown that, by driving at a distance from 5 m to 8 m, all vehicles in the platoon save fuel. In particular, follower vehicles reduce their fuel consumption by up to 16 %, while the leader by up to 8 %.

This shows how important it is to reduce inter-vehicle spacings, but it must clearly be done by a computer-aided driving system, because humans cannot violate the safety distance constraint without increasing the risk of collisions. Close following will thus be made possible by the deployment of an Intelligent Transportation System (ITS), i.e., the umbrella of standards, applications, and technologies that will make transportation systems smarter, more efficient, and more sustainable. One step towards a safer and more efficient driving is the introduction of the Adaptive Cruise Control (ACC) on modern cars. This system automatically maintains a constant cruising speed and a safe distance from any vehicle in front by means of a radar. The ACC, however, keeps a safety distance comparable to the one kept by human drivers, and hence it is not a valid option for improving road capacity.

A fundamental component to realize traffic efficiency and safety applications for an ITS is Inter-Vehicle Communication (IVC), also known with the name of Vehicular Ad Hoc Networks (VANETs), which enables vehicles to communicate between each other or with a centralized infrastructure to obtain and share data on traffic, safety, or other generic information. Since the proposal of IVC, the research community proposed hundreds of applications spanning from traffic efficiency, safety, and infotainment (see examples in [47, 79, 105]). The envisioned application that deals with traffic congestion, fuel saving, and safety is called *platooning*. The idea of platooning is to organize vehicles in groups, called platoons, where the leading one is driving (either autonomously [104] or driven by a professional driver [16]), while the others autonomously follow at a small gap. During standard cruising, no action is required by the driver, i.e., the vehicle accelerates, brakes, and steers autonomously. Besides traffic flow improvement and reduction of fuel consumption, this application has two other advantages. The first one is increased safety: If we consider that the majority of road accidents are due to human errors [25] and we assume that a mechanical fault is less likely to occur, then having an automated system that drives vehicles on a road highly reduces the chances of crashes. The second one is an improved, less stressful driving experience: With such an application, while the system is driving the car, the driver can relax and perform other tasks. For example, if we think about commuters, while driving to work it would be possible to read emails or the newspaper, or drink a cup of coffee. A report for the city of London has shown that 20 % of commuters spend more than two hours traveling to and from work. This includes all transportation means, but it gives an idea on the impact of commuting in our everyday life. Platooning can reduce the amount of time wasted in driving.

The platooning application is composed of different parts. First, we have the control system, i.e., the system that autonomously drives the vehicle. In turn, this system is divided into longitudinal and lateral control. The longitudinal system takes care of accelerating and braking the car (Section 2.2), while lateral control takes care of steering. In this thesis, we focus on the longitudinal component only.

The second fundamental building block is IVC, which is required by the longitudinal control and for managing the platoon. Communication is essential for longitudinal control to enable “safe tailgating”. With sensor-based systems like the ACC it is not possible to reduce the inter-vehicle gap to a value small enough for platooning. Vehicles hence need to share real time data about their dynamics to improve system’s reactivity and enable car-following at close distance. Management is instead needed to create, merge, split, and disrupt platoons, and this can only be performed by means of wireless communication, either direct car-to-car, or with an infrastructure.

The remaining platooning components are more high-level, and include monitoring and advises by a central entity that should further improve the efficiency. The centralized entity might, for example, suggest which platoon to join, or at which speed the platoon should travel to maximize the efficiency. Another aspect that the central entity should manage is billing: Platooning might indeed need a revenue scheme to be socially acceptable and sustainable. In the philosophy of the SARTRE project [16], the leading vehicle is basically driving all vehicles in the platoon, and it is saving less fuel than its followers [66]. The centralized administrative entity should thus provide incentives to platoon leaders, for example by imposing some fees to followers, which must clearly be lower than the amount of money they save thanks to platooning itself.

In this thesis we focus on communication issues concerning longitudinal control. Small scale Field Operational Tests (FOTs) have shown that longitudinal control is feasible [77,84, 88], but what has not been considered so far is the performance of wireless communication systems in large scale scenarios. The longitudinal controller indeed requires frequent and timely updates to safely work, and we must understand if and how we can support platooning in crowded highways. The questions that this thesis addresses are the following:

- Q1: How can we simulate and evaluate the performance of a longitudinal control system for platooning considering a large scale scenario, realistic network communication, and realistic vehicle dynamics?
- Q2: To which extent can the communication technologies proposed for IVC support a platooning control system in a high density scenario?
- Q3: Can we develop a communication protocol capable of delivering timely information to the system?
- Q4: Can we design a control system that can be re-configured at run-time to match network characteristics?

To answer these questions, the thesis is divided in different chapters. In Chapter 2 we describe the fundamental concepts needed to understand the work in this thesis, together with the state of the art background. We give an overview of vehicular communication including communication technologies, standards, applications, and protocols. Moreover, we introduce control systems in general and then focus on longitudinal control for vehicles. Finally, we describe the models required for a realistic simulation of vehicle dynamics for the purposes of this thesis.

Chapter 3 answers question Q1. The chapter presents the platooning simulation framework we developed, including the necessary changes to the base framework we developed on, the structure and the components of the simulator, and some use cases with experimental results.

In Chapter 4 we answer questions Q2 and Q3 and develop a set of protocols that can support platooning control even in highly congested scenarios. The proposed protocols are evaluated through simulations against two well-known approaches in the literature.

Chapter 5 answers question Q4: In the chapter we present and evaluate a longitudinal platooning controller that, in contrast with approaches in the literature, is re-configurable at run-time to work with a chosen communication pattern.

Finally, Chapter 6 presents the conclusion and provides an outlook with future research directions on this topic.

Chapter 2

Fundamentals and Background

In this chapter we describe the fundamental concepts that are needed to understand the work in this thesis and we review the related work on the topic. We start by introducing vehicular communication and the state of the art in information dissemination protocols, with a particular focus on platooning (Section 2.1). Then, we briefly describe control systems, specifically targeting cruise controllers for vehicles (Section 2.2). Further, we describe the concepts behind realistic modeling of vehicle dynamics, detailing mathematical formulation for engine and braking dynamics (Section 2.3). Finally, we review the literature on simulation tools for platooning systems, highlighting features that are considered, required, and lacking as well (Section 2.4).

2.1 Vehicular Communication

2.1.1 Communication Technologies, Standards, and Applications

Inter-Vehicle Communication (IVC) can be realized through different wireless technologies including standard cellular (Universal Mobile Telecommunications System (UMTS) or Long Term Evolution (LTE)), short-range wireless, Visible Light Communication (VLC) [24, 114], etc. Different technologies have different characteristics, advantages, and drawbacks, and each one might be better suited for a particular application. For example, the cellular network can easily support Traffic Information Systems (TISs), i.e., all those applications that are dedicated to the dissemination of traffic information, such as jammed roads due to rush hours or accidents. Indeed, the cellular infrastructure is already deployed and completely covers large cities and freeways, and it would thus be "ready to be used".

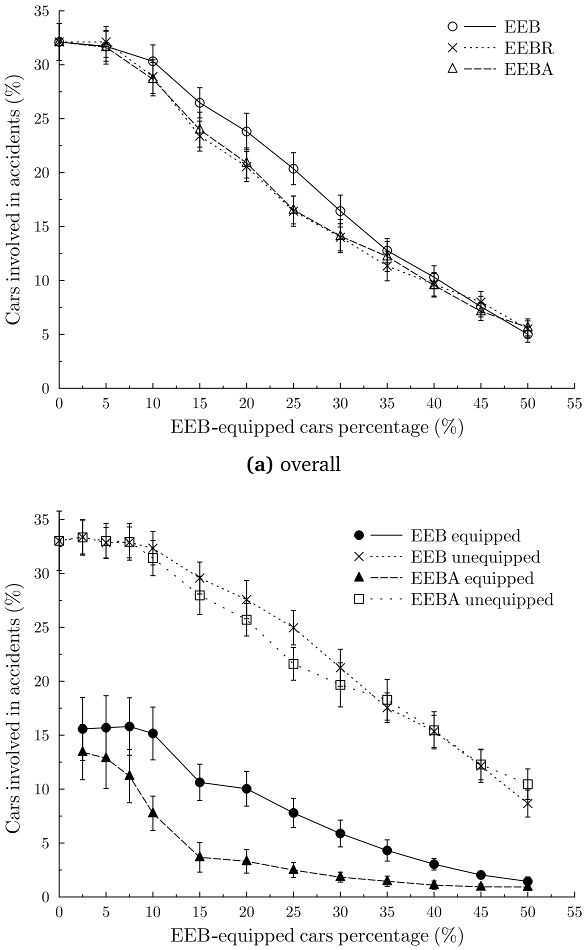
To support safety applications, instead, short range communication is more suited because, in general, safety related information are useful only in the proximity of the event. If a car crashes on a freeway, informing vehicles closer than 1 km is enough to alert drivers and avoid possible further collisions. A technology such as WiFi, or more formally IEEE 802.11 [3], seems fairly suited to the purpose. Indeed, as stated in [63], several studies considered WiFi as a communication technology for vehicular networks [17,21,31,49,126]. Some WiFi drawbacks, however, makes it actually unusable in real world environments. The first problem is that WiFi works in the Industrial, Scientific and Medical (ISM) band: IEEE 802.11b/g/n/ac standards work in the 2.4 GHz band, while IEEE 802.11a/n/ac work in the 5 GHz band. Working in the ISM band means competing for channel access with other technologies like Bluetooth, ZigBee, Near Field Communication (NFC), cordless phones, as well as other 802.11-based applications. This is clearly undesirable, especially when considering life-saving applications. The second problem is given by multi-path propagation. When a signal is sent by a radio device via its antenna, it reaches the receiver via multiple paths by bouncing on objects in the communication environment. The amount of time between the first (direct line of sight) and the last multipath component arrived at the receiver is called the delay spread. In a harsh environment such as the vehicular one, the delay spread is larger than in indoor conditions. For this reason, the standard WiFi protection against Inter Symbol Interference (ISI), called the Guard Interval (GI), is not enough. Third, IEEE 802.11 requires a station to be associated to an access point for communicating, and all the traffic needs to go through the access point, limiting the possibility of having direct car-to-car communication, which is fundamental for safety applications.

The first step towards a dedicated standard was taken in 1999, when the Federal Communications Commission (FCC) reserved the 5.9 GHz band for Dedicated Short Range Communications (DSRC): 7 (in the U.S.) and 5 (in Europe) channels have been defined. Moreover, the IEEE started the standardization of IEEE 802.11p [1]. The aim of IEEE 802.11p is solving the aforementioned issues in IEEE 802.11 without the need of re-designing physical and medium access control layers. Indeed, the standard is based on an Orthogonal Frequency Division Multiplexing (OFDM) PHY and an IEEE 802.11e MAC, so it uses multiple access categories to prioritize traffic. The multi-path propagation issue is mitigated by doubling the OFDM symbol time, and consequently the GI [61]. The bandwidth thus shrinks from the 20 MHz of 802.11a to 10 MHz, so the available datarates range from 3 Mbit/s to 27 Mbit/s. Finally, the association problem is addressed by defining a wildcard Basic Service Set (BSS), (in practice a broadcast MAC address), allowing vehicles to send and receive frames without the need of being associated to an access point [61].

Layers above PHY and MAC are managed by a set of different standards. In the U.S. the IEEE developed the 1609 standard, known with the common name of Wireless Access in Vehicular Environments (WAVE). It is divided in different sub-standards, i.e., 1609.0, 1609.1, 1609.2, 1609.3, and 1609.4, 1609.11, and 1609.12 [52–58]. The aim of IEEE 1609 is to defined higher layer packets format, security primitives, multi-channel operation, etc. In Europe, instead, the ETSI developed the ITS-G5 stack [32, 34]. Similarly to IEEE 1609, ITS-G5 defines message format, channel operations, traffic categories, etc. The standards define two types of messages, Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs). CAMs are single-hop beacons that a vehicle periodically sends to inform neighbors about its presence. In the WAVE standard, the same kind of message is defined as Basic Safety Message (BSM). DENMs, instead, are event-triggered beacons used to inform vehicles about a specific event such as, for example, an emergency braking. To cope with wireless channel congestion, the ETSI standard defines the Decentralized Congestion Control (DCC) algorithm [32–34] (described in details in Section 2.1.2.1). The aim of both North American and European standards is to provide high level vehicular applications with some primitives used to share and obtain data from other vehicles, or from the infrastructure. Both standards are capable of assigning priorities to packets generated by different applications to give more importance, for example, to safety systems.

Since the beginning of the development of IEEE 802.11p, the National Highway Traffic Safety Administration (NHTSA) proposed several different safety applications [79]. On example is the Emergency Electronic Brake Light (EEBL): This application deals with driver's line of sight obstruction. When a vehicle brakes, its brake lights turn on to inform following drivers. If the line of sight is obstructed by, for example, a large truck, the followers will not see the braking signal compromising their possibility of anticipating the action, and may thus lead to a chain collision. By using IVC, the vehicle performing the emergency braking can inform the others, substituting the conventional braking lights with their electronic counterpart.

We investigated EEBL systems from a network and an application perspective [100, 101], showing how effective EEBL is in reducing the chances of a collision, even at low penetration rates. In particular, we simulated a 5-lane freeway scenario with vehicles driven by the Intelligent Driver Model (IDM) [117] and equipped with a DSRC radio and the EEBL application. The leading vehicles performed an emergency braking informing the followers about the danger and, for different market penetration rates, we measured the amount



(a) overall

(b) split between equipped and non-equipped

Figure 2.1 – Car accidents reduction thanks to the EEBL application, as a function of the market penetration rate [101]. Copyright © 2013 IEEE.

of vehicles that collided. Figure 2.1a shows the results for three different dissemination protocols tested. The protocols, named EEB, EEBR (EEB with rebroadcast), and EEBA (EEB with aggregation), are three different flavors of the same application considering different information dissemination protocols. Even for low market penetration rates (15 % to 20 %) EEBL can reduce the number of vehicles involved in the accident by 10 %. With one car equipped every three the crashing vehicles are halved. Moreover, as shown in Figure 2.1b, the vehicles with a DSRC radio are the ones that benefit more, but even non-equipped vehicles have a gain. Indeed, the anticipated and smoother braking maneuver performed by EEBL cars causes the followers to slowly decelerate as well, reducing the chance of a collision. Another result highlighted by the plots is the importance of a correct design of the information dissemination scheme, as this has a non-marginal impact on the safety of the system.

Another example is the Intersection Collision Warning System (ICWS) [62, 64, 65]. With this system the driver can get notified if, while approaching an intersection, another vehicle is in a collision course. If this is the case, the driver receives a warning from his/her own car and can perform an action to avoid a potential crash. Other versions of this application implement an automated reaction [42], so the system decides the car that needs to brake and the one that should accelerate in order to avoid the collision and automatically performs the actions.

Each application clearly has different networking requirements. In ICWS, data must be shared when approaching an intersection, giving higher priority to endangered vehicles [62]. For EEBL, instead, re-propagation improves effectiveness [101]. The platooning application, as a further example, needs a constant, periodic source of data [77, 84]. All applications, still, will need to cope with channel congestion: Sending beacons without checking the network status can have consequences on the performance of the network itself. There is thus the need for a sophisticated access control mechanism that monitors the channel and limits resource utilization, either by reducing beacon rate or by sending beacons only when really required by the application. In the following sections we describe the approaches found in the literature.

2.1.2 Congestion Control in Vehicular Networks

A still active field of research in vehicular networks is the development of communication and information dissemination protocols that can cope (and prevent) channel congestion. Protocols that are channel-unaware can indeed saturate the network, which in turn might harm the applications. For this reason, in the scientific literature and standardization we find several different approaches that cope with channel congestion control. The main idea is to monitor the (potential) use of the channel through different metrics, which include for example:

- Channel busy ratio: This is the amount of time the physical layer declares the channel busy over a certain time window, e.g., 1 s;

- Neighbors: By counting the number of different vehicles we received a beacon from, we can estimate how dense the network is;

- Packet error rate: This can be estimated by checking the sequence numbers included in beacons from nearby vehicles.

Then, depending on such metrics, we can adapt the following parameters:

- Packet generation rate: How fast we try to inject beacons into the network;
- Transmit power: Reducing the power reduces interference caused to farther vehicles;
- Modulation and coding scheme: Physical layer modulations like Binary Phase-Shift Keying (BPSK) or Quadrature Phase-Shift Keying (QPSK) are more robust than Quadrature Amplitude Modulation (QAM), i.e., for the same amount of interference it is more likely for a receiver to successfully decode a BPSK- or QPSK-encoded frame than one encoded with QAM. However, using BPSK and QPSK will result in longer channel utilization time because for those modulations the physical layer bit rate is lower;
- Clear Channel Assessment (CCA) threshold: This threshold is used to define the amount of energy needed at the PHY layer to declared the channel as busy when the preamble portion of a frame is missed, which might occur because a station was transmitting or due to cumulative interference. Changing this threshold affects how sensitive a station is to far vehicles.

One of the contributions of this thesis is the development of safe and efficient beaconing algorithms for a platooning application. Our proposed approaches are compared against two protocols available in the literature. The first is the DCC algorithm proposed by the ETSI as a standard for vehicular communication in Europe. The second one is Dynamic Beaconing (DynB), a protocol we also contributed to develop. As we use these protocols for comparison, we explain the concept behind both protocols in two dedicated sections Sections 2.1.2.1 and 2.1.2.2.

Besides DCC and DynB, in the literature we can find different approaches to tackle the problem of channel congestion in vehicular networks. For example, LInear MEssage Rate Integrated Control (LIMERIC) [12] is a control-theoretic protocol that makes the channel load converge to a desired value. Basically, it monitors the channel load over time and reduces/increases the beaconing rate if the load is higher/lower than the desired one. The control formula computes the beacon rate as

$$r_j(t) = (1-\alpha)\, r_j\,(t-1) + \beta\left(r_g - r\,(t-1)\right), \tag{2.1}$$

where r_g is the “goal” rate, i.e., the beacon rate all vehicles wish to converge to, while $r(t-1)$ is the sensed network capacity being in use, i.e., $r(t-1) = \sum_{i=1}^{K} r_j(t-1)$, with K being the (unknown) number of vehicles. The α and β parameters can be configured to control fairness, stability, and steady state convergence properties of the algorithm and their value must be bounded:

$$0 < \alpha < 1, \qquad \beta > 0. \tag{2.2}$$

The authors formally derive stability conditions for the algorithm. In particular, they prove that if the following conditions holds

$$1 - \alpha - K\beta > -1, \quad \alpha + K\beta < 2, \tag{2.3}$$

then all vehicles will converge to the same beacon rate

$$r_j = \frac{\beta r_g}{\alpha + K\beta} \tag{2.4}$$

and that the total channel load converges to

$$r = \frac{K\beta r_g}{\alpha + K\beta}. \tag{2.5}$$

As stability conditions of Equation (2.3) include the unknown parameter K, α and β needs to be chosen depending on an envisioned value of K. For further details on this algorithm, the reader can refer to [12, 14, 67].

A different way of controlling the load is by adapting the transmit power. Torrent et al. [116] develop the Distributed Fair Power Adjustment for Vehicular environments (D-FPAV) protocol, which solves an optimization algorithm in a distributed way. The algorithm maximizes the transmit power under a maximum channel load constraint. Their approach does not only account for CAMs, but also for sporadic DENMs.

LIMERIC and D-FPAV try to keep the channel load at a desired level, in order to give a minimum guarantee to all applications. However, they give no preference to particular applications, i.e., they are application unaware. Other works thus focus on application layer for adapting beacon rate and/or transmit power. One example is the work by Huang et al. [48]. Their approach use a model predictive mechanism, i.e., based on network statistics each vehicle predicts the error between its own actual position and the position that might have been computed by the others. This is called the Suspected Tracking Error

(STE). By using the STE, the vehicle computes the probability of transmission for the next time slot. In addition, the algorithm adapts the transmit power based on currently sensed channel load. The idea is that the channel load is correlated to vehicle density: A higher load should be caused by a higher density of vehicles. To get the load under control, the algorithm lowers the transmit power but, being in high density conditions, the vehicles will still be able to reach the closest neighbors, which are the most important from a safety point of view. Conversely, high power is used when the density is lower, thus the vehicles can reach farther ones. The algorithm is compared to standard 10 Hz and 2 Hz beaconing: The approach shows better tracking accuracy for nearby vehicles. Tracking accuracy gets worser as the distance increases but, from a safety point of view, this is not critical.

Another work using the STE concept is the one by Bansal et al. [13]. The authors propose the Error Model based Adaptive Rate Control (EMBARC) algorithm, which modifies LIMERIC to account not only for channel load, but also for STE when computing the beacon rate.

Zemouri et al., propose a different way of solving the problem [127]. Their approach explicitly takes into account a minimum update requirement, as well as channel busy ratio and packet losses. Upon these parameters, the algorithm reduces the beacon rate trying to converge to a target value for busy time and collisions. If the minimum update requirement is not met, the algorithm then adapts the transmit power until a reaching a satisfying awareness level among neighbors.

Similarly to this approach, Sepulcre et al. develop a dynamic protocol in which each application can tell the protocol its own update requirement [103]. The requirements are then taken into account when computing the beacon rate.

Zhang et al. [128] propose a distributed optimization that adapts the probability of transmission of a slotted p-persistent approach (originally proposed in [115]). The optimization problem is based on a utility function that takes into account both the safety benefit and the expected delay.

There exist other kind of protocols that focus on specific applications. In this thesis we explicitly consider approaches for the Cooperative Adaptive Cruise Control (CACC) application, i.e., the one that automatically controls vehicles in a platoon using data shared wirelessly among the cars (see Section 2.2.5 for more details). As witnessed in the literature, indeed, packet losses can have dramatic effects on safety and stability of the platoon [73, 82]. For this reason we find several studies that focus on message dissemination for a CACC. One example is the work by Fernandes and Nunes [36]. The authors propose and analyze five different protocol alternatives all based on a Time Division Multiple Access (TDMA) concept. As in this thesis, they consider the CACC developed in the PATH project [86, 88] and, to cope with different situation, they dynamically adapt controller parameters as well. Böhm et al. analyze the use of different MAC layer Access Categorys (ACs) for prioritizing CAMs (used for platooning) against DENMs [19]. Tielert et al. [113] investigates the use of adaptive transmit power control for generic safety applications. The authors show that this technique is, in general, not beneficial. In this thesis we try to consider it in the context of platooning.

To improve the performance of the network we can also consider external resources. For example, we might consider to use the infrastructure [10] or multiple channels [45]. A promising approach is the use of Visible Light Communication as a backup technology to enhance the connectivity between successive vehicles in a platoon [9, 102].

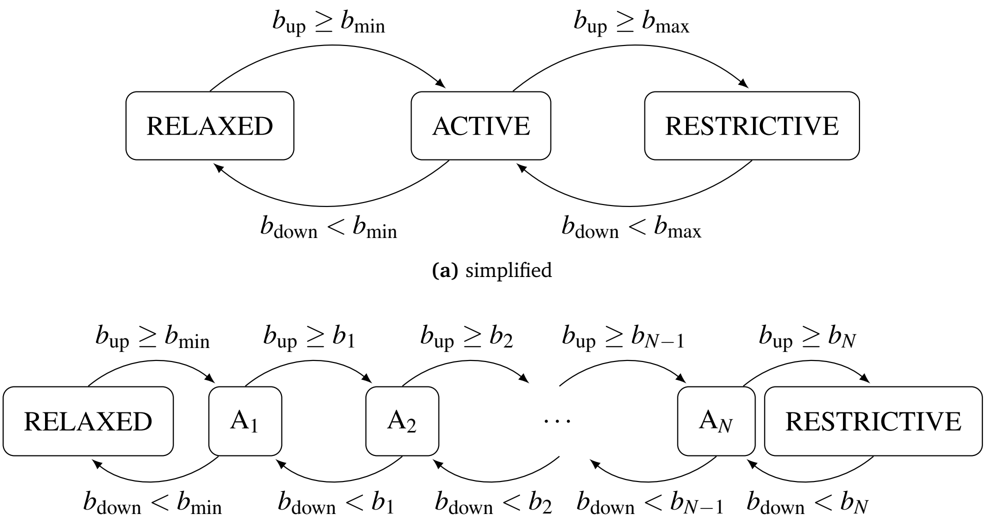
2.1.2.1 ETSI Distributed Congestion Control (DCC)

As briefly mentioned in Section 2.1.1, ETSI, within its ITS-G5 protocol suite, developed DCC. The protocol is defined in various standard documents [32–34] describing CAM generation rules. DCC, to cope with channel congestion, can adapt beacon rate, transmit power, modulation and coding scheme, and CCA-threshold simultaneously. In particular, the adaptation mechanisms defined by DCC are named Transmit Rate Control (TRC), Transmit Power Control (TPC), Transmit Datarate Control (TDC), and DCC Sensitivity Control (DSC). Coupled with channel layer control metrics, the latest release includes a set of vehicle dynamics-based rules for CAM triggering.

The standard defines a state machine, which drives each component of the algorithm. DCC, based on currently observed channel busy ratio, chooses which state to activate. In this thesis we are only interested in the 3-state state machine designed for the Control Channel (CCH), i.e., the channel designated to CAMs [34]. The standard, however, defines rules and parameter for the Service Channel (SCH) as well.

The algorithm periodically samples the busy ratio and keeps two time windows of such samples: T_{down} and T_{up}. At time t, it computes $b_{\text{down}} = \max\left\{b_{t-T_{\text{down}}}, \dots, b_t\right\}$ and $b_{\text{up}} = \min\left\{b_{t-T_{\text{up}}}, \dots, b_t\right\}$, where b_t, $b_{t-T_{\text{down}}}$, and $b_{t-T_{\text{up}}}$ are the channel loads measured at times t, $t - T_{\text{down}}$, and $t - T_{\text{up}}$ respectively. The protocol then performs a state change by comparing these values with thresholds b_{min} and b_{max}. State change is performed according to the following rules:

- If $b_{\text{down}} < b_{\text{min}}$, set the state to RELAXED;
- If $b_{\text{up}} \geq b_{\text{max}}$, set the state to RESTRICTIVE;
- Otherwise set the state to ACTIVE.



(a) simplified

(b) with all ACTIVE sub-states

Figure 2.2 – DCC state machines.

Figure 2.2a shows a graphical representation of the principle of operation. The standard permits the definition of further sub-states within the ACTIVE state. Each of those ACTIVE sub-states i defines a maximum channel load b_i and DCC-related parameters. The states are ordered according to b_i, so that $b_{i-1} < b_i, i = 0, \ldots, N+1$, with N being the number of ACTIVE states, $b_0 = b_{\text{min}}$, and $b_{N+1} = b_{\text{max}}$. In the ACTIVE sub-states, state transitions are performed by finding the state id $i = \max(i_{\text{up}}, i_{\text{down}})$ such that

$$b_{i_{\text{up}}-1} \leq b_{\text{up}} < b_{i_{\text{up}}} \tag{2.6}$$

$$b_{i_{\text{down}}} < b_{\text{down}} \leq b_{i_{\text{down}}+1}. \tag{2.7}$$

Figure 2.2b shows the DCC state machine with all ACTIVE sub-states. For the sake of clarity, we only draw the state changes between consecutive states, but according to the standard the states are fully meshed, and the transitions simply follow the rules in Equations (2.6) and (2.7).

For the CCH, the standard considers a single ACTIVE state. In this thesis, we consider the AC_VI access category for the generation of CAMs. Table 2.1 lists the default parameters defined in the standard [32–34]. A “ref” value indicates that the corresponding parameter is unchanged when switching from the old to the new state. In the standard they are

Table 2.1 – DCC parameters for the control channel, AC_VI. Copyright © 2015 IEEE.

	RELAXED	ACTIVE	RESTRICTIVE
b_i	0.15	0.20	0.40
Tx power	33 dBm	ref	−10 dBm
Packet interval	0.04 s	ref	1 s
Datarate	3 Mbit/s	ref	12 Mbit/s
CCA threshold	−95 dBm	ref	−65 dBm

only provided as an indication, but finding and using a different set of parameters that maximizes DCC performance is out of the scope of this work.

Following the state change rules of Equations (2.6) and (2.7) and configuring DCC with the parameters in Table 2.1, the state machine switches to the RELAXED state for channel loads lower than 15 %, to the ACTIVE state for channel loads between 15 % and 20 %, and to the RESTRICTIVE state for channel loads higher than 20 % (this is confirmed by the example shown in [32, Figure 6]).

In addition to the DCC mechanism, a dedicated standard defines a set of CAM triggering rules, built on top of DCC rules [34]. EN 302 637-2 re-defines minimum and maximum CAM generation intervals, i.e., 0.1 s and 1.0 s, respectively. Moreover, the minimum CAM generation interval is further restricted depending on the current DCC state. In [33] are defined a set of DCC Profiles (DPs) to characterize traffic streams in the access, the network, and the transport layers. Those DPs are numbered from 0 to 32 (0 being the traffic with the highest priority) and each one is associated with a T_{off} parameter that regulates packet interval rules for each DCC state. According to [34], CAMs belong to the DCC profile DP2, thus the minimum interval is restricted to 95 ms, 190 ms and 250 ms for the RELAXED, the ACTIVE, and the RESTRICTIVE states respectively [33, Table 1].

Further, the standard defines vehicle dynamics-based triggering conditions to alert nearby vehicles in case of sudden changes in the state of the vehicle. These conditions can be extended according to new applications' requirements. The default conditions are the following:

- The absolute difference between last sent heading and current heading direction exceeds 4°;
- The distance between last sent position and current position exceeds 4 m;
- The absolute difference between last sent speed and current speed exceeds 0.5 m/s.

If any of the aforementioned conditions is met and the minimum packet interval T_{off} has elapsed, a CAM should be immediately generated and sent. Moreover, if the CAM is triggered due to the dynamics-dependent conditions, the protocol must schedule three consecutive CAMs with an interval equal to the time elapsed since the last CAM generation. The packet interval must be reset to the maximum (i.e., 1 s) when all the repetitions are sent.

In [14], the authors analyze the behavior of DCC. The algorithm is capable of keeping the resource utilization under control, but it shows an unstable behavior due to its state-machine design. Instability is undesirable, as the channel load might oscillate instead of converging to a steady-state value. This might lead to changes in the design of DCC in future standard releases.

2.1.2.2 Dynamic Beaconing (DynB)

In [109, 110] we contributed to the proposal of the DynB protocol, which, similarly to LIMERIC, uses a control-theoretic approach to maintain the channel load under control. DynB differs from ETSI DCC in the following characteristics:

- It only adapts the beacon interval and does not act on PHY layer parameters;
- It is thought for TIS applications, so it does not consider vehicle dynamics;
- It computes the beacon interval by measuring the channel busy ratio and the number of neighboring vehicles;
- The channel load converges to a predefined value.

The DynB control formula is defined as

$$I = I_{\text{des}}\left(1 + rN\right), \tag{2.8}$$

where I is the computed beacon interval, I_{des} the desired beacon interval, i.e., the minimum interval that a vehicle can use. The r term, instead, is defined as follows:

$$r = \max\left(0, \min\left(1, \frac{b_t}{b_{\text{des}}} - 1\right)\right). \tag{2.9}$$

In Equation (2.9), b_t and b_{des} are the measured and desired channel load respectively. In [110] we suggest a b_{des} value of 25 % and we show how fast DynB reacts compared to the TRC mechanism of DCC. Thanks to this, the algorithm can take advantage of fast changing channel conditions caused by time-varying shadowing. The downside of DynB is that the beacon interval has to increase with the number of vehicles to maintain the desired channel load. As we show in Chapter 4, this has a negative impact on awareness.

2.2 Control Systems

A control system regulates or supervises other systems by acting on some inputs in order to achieve a desired goal. The simplest example is the automatic regulation of the temperature in a room. The goal is to reach and maintain the temperature chosen by a human, and the system is the heating (or cooling) system. The control system measures the current room temperature and compares it with the required one: If the former is lower than required, the controller tells the heating system to warm up the room, while if it is too high the room must be cooled down. Control systems are used to perform an enormous amount of tasks in our everyday life, from regulating the temperature of a room to automatically fly or land an aircraft. In this thesis, the goal is to longitudinally control a platoon of vehicles maintaining a desired speed or inter-vehicle gap. Another fundamental component of platooning is the lateral steering control, which, however, is out of scope for this thesis.

Figure 2.3 shows a schematic representation of a control system: r is the reference signal (i.e., the goal) which, summed to the measured signal y_m, produces an error e. The controller $C(s)$, based on such error, computes a control input u which is then fed, with some appropriate means, into to the plant $P(s)$, i.e., the actual system. Depending on u and the external disturbances, the system changes its current state, resulting in a control output y.

To better understand this concept, we anticipate the content of Section 2.2.3 and take as an example a Cruise Control (CC). A CC is a control system available on modern cars that automatically maintains a desired speed. In this case, the reference signal r is the desired speed, and the error e is the difference between r and the speed y_m measured by the speedometer. The control input u is a signal that tells the engine to speed up or to slow down: For example, it might tell the engine control unit to actuate a particular throttle position. The disturbances are the external frictions such as air drag, rolling resistance, and the gravity pulling the car while climbing a hill. The output y will be the actual state of the car, so its position, speed, acceleration, and any other quantity that might be needed to describe it. In the following sections we describe in details the longitudinal control systems we consider in this thesis, as well as the modeling required to realistically simulate the dynamics of a vehicle.

2.2.1 Actuation Lag

Before describing longitudinal control systems we need to describe actuation, i.e., when the plant performs what the controller has computed. This actuation might be delayed

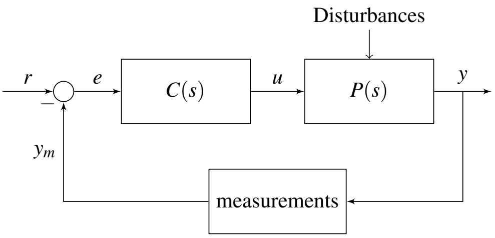


Figure 2.3 – A simple feedback control system.

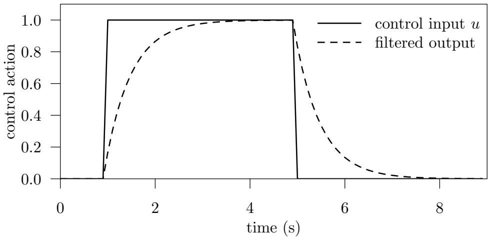


Figure 2.4 – First order lag applied to control input u for $\tau = 0.5\,\mathrm{s}$.

due to the internal plant dynamics. In a vehicle, for example, when the controller tells the car to accelerate it needs to send the input u to the engine control unit, which then opens fuel valves to accelerate the crankshaft that, in the end, accelerates the car. This process is clearly not immediate and is referred to as *actuation lag* [84, 86].

In the analysis of longitudinal control system for vehicles, this acceleration lag is usually modeled as a first order lag (i.e., using a first order low-pass filter) with a time constant τ around 0.5 s [84, 86, 90]. This time constant is large but it accounts for the response of the engine, delay of sensors, sampling delay, etc., and can be seen as a worst-case delay. Being 0.5 s the worst case, any vehicle can be designed to respond with such delay [84].

To implement a first order lag in a discrete simulator we use the following formula:

$$\ddot{x}[k] = \begin{cases} u[0] & \text{if} \quad k = 0 \\ \alpha \cdot u[k] + (1-\alpha) \cdot \ddot{x}[k-1] & \text{otherwise} \end{cases} \tag{2.10}$$

where $\ddot{x}[k]$ is the acceleration of the vehicle at time step k and $u[k]$ is the control input. For simplicity, in this thesis we assume that the control input u is the desired acceleration. The parameter α is a constant that depends on the simulation sampling time Δ_t and the time constant τ and is computed as

$$\alpha = \frac{\Delta_t}{\tau + \Delta_t}. \tag{2.11}$$

Figure 2.4 shows the step response of the first order lag for a value of $\tau = 0.5\,\mathrm{s}$. The solid line represents the desired acceleration u, while the dashed one the actual acceleration $\ddot{x}$.

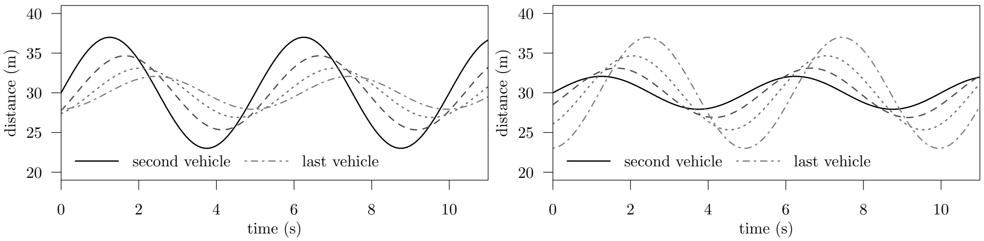
The actuation lag can be also modeled more realistically, for example by taking into account engine and brakes characteristics. Equation (2.10) assumes that engine and brakes actuation have the same dynamics, which is not the case in reality. In Section 2.3 we describe a more realistic model for engine and brakes actuation.

2.2.2 String Stability

A fundamental concept in the analysis of control algorithms for automated car following is the concept of string-stability. A platooning control algorithm is said to be string-stable if any error in position, speed, or acceleration by a vehicle is not amplified towards the end of the platoon [20, 111]. This is a different concept from the classical stability notion of a control system that, for example, concerns only a single vehicle. String stability is a property of the platoon as a whole and it is of utmost importance because it ensures that, under normal conditions, the system does not cause vehicle collisions. The standard approach for proving the string-stability of a controller is to analyze its response to a sinusoidal input. If the amplitude of such a sinusoid is not amplified by vehicles at the back, then the controller is string-stable. For a graphical explanation consider Figure 2.5, which shows artificially generated distance traces of a stable (Figure 2.5a) and an unstable (Figure 2.5b) controller. In the first case, the oscillation of the distance between the first and the second vehicle is attenuated by following vehicles. In the second case, instead, the oscillation amplitude between the last and the second to last vehicle is greater than the one of vehicles in front. This instability can lead to a crash for a larger number of vehicles in the platoon or for a stronger oscillation at the head. For this reason, string stability is a required property for any platooning controller.

2.2.3 Cruise Control

As briefly mentioned in the previous sections, a CC is a control system that lets the driver choose a desired speed that is automatically maintained by the car. A CC does not perform automatic braking: If the vehicle approaches a slower one in front, the driver needs to



(a) string stable

(b) string unstable

Figure 2.5 – Artificially generated inter-vehicle distance traces showing string stable and string unstable behavior.

manually disengage the controller. Still, such a system can enhance driving comfort during long freeway trips.

The control law for a CC is defined as [86, Chapter 5]

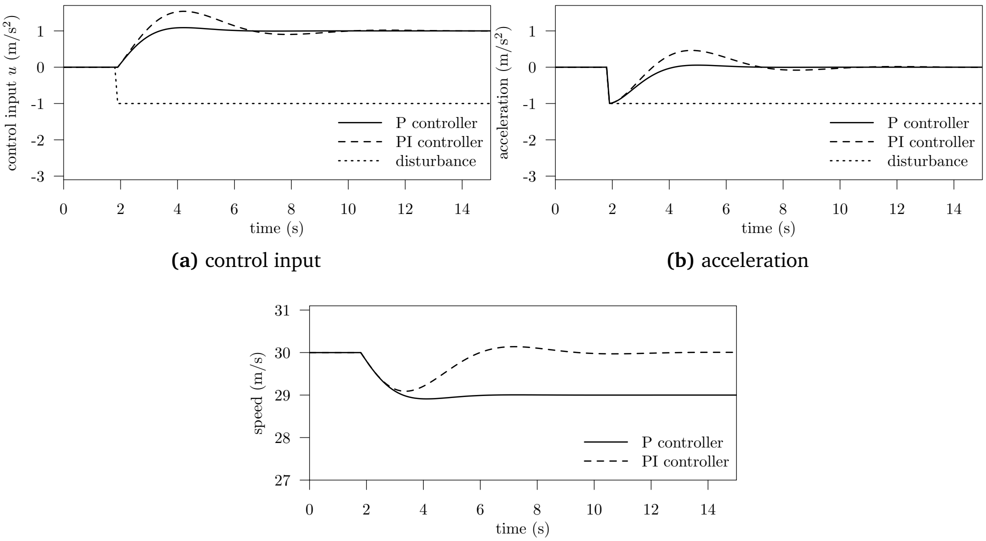
$$u = -k_p\left(\dot{x} - \dot{x}_d\right) - k_i \int \dot{x} - \dot{x}_d dt, \tag{2.12}$$

where $\dot{x}$ and $\dot{x}_d$ are the current and the desired speed respectively, while x and x_d are the positions the car would have by traveling at those speeds. This kind of controller is known as Proportional Integral (PI) controller, because the law computes the control action using both a proportional and an integral component of the error. The parameters k_p and k_i can be changed to tune the behavior of the controller. By setting $k_i = 0$, the controller reduces to a simple proportional controller.

The PI design makes the controller robust to external disturbances. Figure 2.6 shows the behavior of a proportional and a proportional integral CC configured to maintain a desired speed of 30 m/s. At time $t = 2$ s an external disturbance of 1 m/s^2 is applied. Both controllers react to the disturbance by increasing their control input to compensate the disturbance, but only the PI CC is capable of stabilizing the speed to the desired speed of 30 m/s (Figure 2.6c). For the proportional controller, instead, the speed converges to 29 m/s, thus never reaching back the reference value chosen by the driver.

2.2.4 Adaptive Cruise Control

A more sophisticated version of the CC is the Adaptive Cruise Control (ACC). The ACC uses a radar or a lidar to detect and monitor vehicles in front. The user chooses a desired speed as with the CC. If the radar detects a slower vehicle ahead, the ACC automatically



(a) control input

(b) acceleration

(c) speed

Figure 2.6 – Comparison of a proportional and a proportional integral CC subject to an external disturbance. In this plot, $\dot{x}_d = 30\,\mathrm{m/s}$, $\tau = 0.5\,\mathrm{s}$, $k_p = 1$, and $k_i = 0.5$ (for the PI controller).

decelerates and maintains a safe gap between the own and the front vehicle. If the road ahead becomes free again, for example when changing lane to overtake, the ACC accelerates again and brings the car to the reference speed chosen by the driver.

The control law for the ACC is defined as [86, Chapter 6]

$$u_i = -\frac{1}{T}\left(\dot{x}_i - \dot{x}_{i-1} + \lambda\left(x_i - x_{i-1} + l_{i-1} + T\dot{x}_i\right)\right), \tag{2.13}$$

where $\dot{x}_i$ and $\dot{x}_{i-1}$ are the speed of the considered and the front vehicle respectively, x_i and x_{i-1} are their positions, l_{i-1} is the length of the front vehicle, T is the headway time, and λ is a design parameter. The radar computes the relative speed $(\dot{x}_i - \dot{x}_{i-1})$ and the distance to the front vehicle $(x_i - x_{i-1} + l_{i-1})$. The desired distance $T\dot{x}_i$ is not fixed, but depends on the cruising speed. In particular, the ACC maintains a distance of T seconds from the front vehicle, so the actual distance is constant in time. This spacing policy is called constant time-gap policy, and it is required to guarantee the string stability properties of the system. Moreover, T cannot be chosen arbitrarily: in particular, as proven in [86, Chapter 6], string stability is guaranteed only if the following holds:

$$T \geq 2\tau, \tag{2.14}$$

i.e., the time headway must be at least twice as large as the actuation lag. For an actuation lag of 0.5 s, this means that the time gap must be larger than 1 s.

2.2.5 Cooperative Adaptive Cruise Control

Because of the large spacing the standard ACC maintains, it is clear that it is not suitable for accomplishing the following platooning goals: Increasing the road throughput and reducing fuel consumption. For this reason the research community started to work on an enhanced version, which exploits wireless communication among vehicles. The idea is to share information such as acceleration, speed, position, etc., to improve the reactivity of the system, and because vehicles cooperate by sharing such information, this enhanced version of the ACC has been named CACC.

The literature reports several different CACCs which differ in design, characteristics, and requirements [11, 77, 84, 86, 90]. Here, we give a quick overview of some of them, but we only detail the ones that have been selected and used in the rest of th thesis.

CACCs designs mainly differ for the so called *control topology* [90] that indicates which vehicles' data the controller considers. In the case of a standard ACC, the radar can only provide information about the vehicle directly in front. By using wireless communication, instead, each vehicle can exploit data of two vehicles in the front, or of the leader, or even of vehicles behind. A first example resembling the standard ACC is described in [84]. This CACC employs only the data of the vehicle directly in front. In particular, it obtains distance and relative speed from the radar as the ACC does, but it also exploits the desired acceleration (i.e., the control input u) received via the radio interface. Sharing and using the desired acceleration instead of the actual one gives an advantage in terms of system reactivity, because each vehicle tells its follower a future information about what it will be shortly doing: Clearly, this information cannot be measured by any sensor.

The control formula for this CACC is defined as

$$\dot{u}_i = \frac{1}{T}\left(-u_i + k_p\left(x_{i-1} - x_i - l_{i-1} - T\dot{x}_i\right) + k_d\left(\dot{x}_{i-1} - \dot{x}_i - T\ddot{x}_i\right) + u_{i-1},\right) \tag{2.15}$$

where T is the time headway as in Equation (2.13), k_p and k_d are gains used to tune the behavior of the controller, and u_{i-1} is the desired acceleration of the front vehicle received by means of wireless communication. As for the ACC, the string stability of this controller is guaranteed in the presence of a constant time-gap policy. The time headway T, however, can be reduced down to 0.5 s [84], so twice as small as for the ACC.

The authors also thoroughly investigated the impact of the network on the stability of the controller, computing different headway times for different achievable packet inter-arrival times [82]. Moreover, they show that in complete absence of network communication, the desired acceleration u_{i-1} can be substituted with the actual acceleration $\ddot{x}_{i-1}$ computed using the radar and by increasing the time headway T to ensure system stability [85].

In the scope of the California PATH project [104] the researchers developed a controller with a different control topology. In particular, the design of the controller exploits data received by the leader and the vehicle in front. This choice provides string stability of the platoon under a constant spacing policy, i.e., the inter-vehicle distance can be arbitrarily chosen and does not depend on the speed. As an example, in the experimental validation performed in [88], the vehicles maintained a constant gap of 6 m. The SARTRE project [16] adopted a similar concept, testing inter-vehicle distances down to 5 m [66].

The control law for this CACC is defined as [86]

$$u_i = \alpha_1 u_{i-1} + \alpha_2 u_0 + \alpha_3 (x_i - x_{i-1} + l_{i-1} + d_d) + \alpha_4 (\dot{x}_i - \dot{x}_0) + \alpha_5 (\dot{x}_i - \dot{x}_{i-1}), \quad (2.16)$$

where d_d is the desired distance in meters, while α_i are defined as

$$\alpha_1 = 1 - C_1; \quad \alpha_2 = C_1; \quad \alpha_5 = -\omega_n^2 \quad (2.17)$$

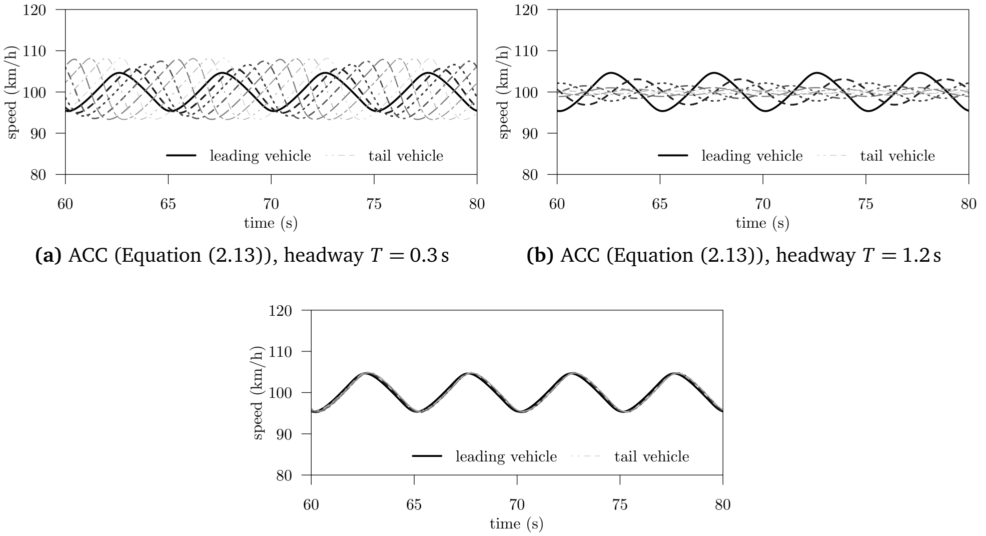
$$\alpha_3 = -\left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)\omega_n \quad (2.18)$$

$$\alpha_4 = -C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n. \quad (2.19)$$

C_1 is a weighting factor between the accelerations of the leader and the preceding vehicle, ξ is the damping ratio, and ω_n is the bandwidth of the controller. These parameters can be changed to tune the behavior of the controller. In Equation (2.16), the inter-vehicle distance is obtained through the radar, while acceleration and speed of leader and front vehicles are received through the radio interface.

This controller is the reference CACC we consider for the majority of the analyses in this thesis. We choose the PATH controller because, due to its constant spacing policy, it gives the highest benefits in terms of road throughput and fuel saving.

Another completely different approach is to consider a configurable control topology. In [90], we develop a controller where each vehicle is potentially capable of exploiting data from every other vehicle in the platoon. Given that the controller is part of our contribution to the scientific community, we described it in detail in Chapter 5.



(a) ACC (Equation (2.13)), headway $T = 0.3\,\mathrm{s}$

(b) ACC (Equation (2.13)), headway $T = 1.2\,\mathrm{s}$

(c) CACC (Equation (2.16))

Figure 2.7 – Speed profiles for an 8-car platoon using ACC and PATH's CACC [99]. Copyright © 2014 IEEE.

To conclude this section, we show a comparison between ACC and the PATH's CACC we will mainly consider in the remainder of the thesis. In [99] we report about a simulation of an 8-car platoon traveling on a freeway at an average speed of 100 km/h. Leader's speed follows a sinusoidal disturbance profile with a frequency of 0.2 Hz and an amplitude of roughly 5 km/h. Follower vehicles are either driven by an ACC or by the PATH's CACC. For the ACC, we consider two configurations, one unstable ($T = 0.3\,\mathrm{s}$) and one stable ($T = 1.2\,\mathrm{s}$). The PATH's CACC, instead, is configured to maintained a fixed, 5 m gap. Figure 2.7 shows the speed of the vehicles in time. The ACC using the unstable configuration results in an amplification of leader's motion, which is uncomfortable and potentially dangerous. In a stable configuration, the ACC behaves in a safe manner, but keeps an average inter-vehicle distance of roughly 33 m. The speed profiles of vehicles using CACC are indistinguishable. Each vehicle almost perfectly reproduce leader's behavior and at a distance of only 5 m. This example shows the potential of an IVC-based CACC compared to standard sensor-based systems.

2.3 Realistic Vehicle Modeling

In Section 2.2.1 we described a simplified engine model using a first order lag. As briefly mentioned, a first order lag assumes acceleration and braking dynamics to be symmetric, and does not consider physical limits. When focusing on network and protocol analysis such assumptions are reasonable, but when we are interested in studying realistic platoon dynamics we need a more accurate model. To properly model vehicle dynamics we need to consider physical laws. For simplicity, vehicular simulations consider a constant value for maximum acceleration, which, in practice, makes the car more reactive that it actually is. In reality we have an engine with a limited power, external forces, mass, gear ratios, etc., and maximum acceleration is a function of such parameters. In this section we thus describe a detailed engine model that we implemented and made available in the PLEXE simulator [99]. We derive fundamental equations for the vehicle dynamics using the generalized Newton's second law and the D'Alembert's principle.

We start by defining the following force balance:

$$\lambda m \ddot{x} = F_i = F_u - F_F, \tag{2.20}$$

with F_i, F_u, and F_F being the inertial, effective traction, and friction forces, respectively, m and $\ddot{x}$ are the mass and the acceleration of the vehicle. The parameter λ is a factor that accounts for the inertia of the rotating mechanical components in the driveline. The friction forces are defined as [38, 68]:

$$F_F = F_A + F_R + F_G, \tag{2.21}$$

where F_A is the air resistance [68], F_R is the rolling resistance [124], and F_G is the gravitational force. In turn, each friction component is defined as:

$$F_A = \frac{1}{2} c_{\text{air}} A_L \rho_a \dot{x}^2, \tag{2.22}$$

$$F_R = mg\left(c_{r1} + c_{r2}\dot{x}^2\right), \tag{2.23}$$

$$F_G = mg\sin\left(\theta_{\text{road}}\right), \tag{2.24}$$

where c_{air} is the air drag coefficient, A_L is the maximum vehicle cross section area, ρ_a is the air density, c_{r1} and c_{r2} are parameters that depend on the tires and their pressure, g is the gravitational acceleration, and θ_{road} the slope of the road expressed in degrees. We can now rewrite the longitudinal motion law using Equation (2.20) and by taking into account the actuation lag:

$$\ddot{x} = \frac{1}{\lambda m}\frac{1}{1+\tau(\dot{x})s}F_u - \frac{1}{\lambda m}F_F, \tag{2.25}$$

where $\tau(\dot{x})$ is the speed-dependent lag during acceleration and braking maneuvers.

F_u can be either a propelling or a braking force according to the control input u imposed by the controller. Propelling force is generated by the engine, while braking is originated by the friction between disks and pads:

$$F_u = \begin{cases} -F_{\text{brake}} & \text{if} \quad u \leq 0 \\ F_{\text{eng}} & \text{if} \quad u > 0. \end{cases} \tag{2.26}$$

F_u is bounded ($F_{u\text{min}} \leq F_u \leq F_{u\text{max}}$) because of engine and brakes limits, which depend from the actual vehicle speed and gear box status ($F_{\text{eng}_{\text{min}}} \leq F_{\text{eng}} \leq F_{\text{eng}_{\text{max}}}$ and $F_{\text{brake}\,\text{min}} \leq F_{\text{brake}} \leq F_{\text{brake}\,\text{max}}$). Furthermore, we need to consider two different actuation lags for the engine and for the braking system:

$$\tau(\dot{x}) = \begin{cases} \tau_{\text{brakes}}(\dot{x}) & \text{if} \quad u \leq 0 \\ \tau_{\text{eng}}(\dot{x}) & \text{if} \quad u > 0. \end{cases} \tag{2.27}$$

For the sake of clarity, we derive engine and brakes bounds and lags in two separate sections.

2.3.1 Engine Acceleration and Lag

Maximum engine force depends on its torque and power. In particular, the traction F_{eng} depends on the engine power P_{eng} which, in turn, is a function of the current engine speed [68]:

$$F_{\text{eng}} = \frac{\eta P_{\text{eng}}(N_{\text{eng}})}{\dot{x}} \qquad [\text{N}], \tag{2.28}$$

where P_{eng} expressed in [W], N_{eng} is the engine speed in [rpm], η is the engine efficiency, and $\dot{x}$ is the speed in [m/s]. We can easily find engine power and torque curves online, as a results of the dyno tests performed by manufacturers or by other institutions.[1]

[1] http://rototest-research.eu/

As the wheels are directly connected to the engine via the differential and the gearbox, we can link engine and vehicle speed by [38]:

$$N_{\text{eng}} = \frac{60 i_d i_g \dot{x}}{d_{\text{wheel}} \pi} \qquad [\text{rpm}]. \tag{2.29}$$

In Equation (2.29), i_d and i_g are the differential and the currently engaged gear ratio, and d_{wheel} is the diameter of the tractive wheels in [m]. By substituting Equation (2.29) into Equation (2.28) we can derive F_{eng} for all possible gear ratios.

If instead of the power curve we have the engine torque curve, we can obtain the force at the wheel using a slightly different formulation. The relationship between power and torque is

$$P(N_{\text{eng}}) = T(N_{\text{eng}}) \cdot \omega = T(N_{\text{eng}}) \cdot \frac{2\pi}{60} N_{\text{eng}} \qquad [\text{W}], \tag{2.30}$$

where T is the engine torque in [N m] and ω is the engine angular speed in [rad/s]. Substituting Equations (2.29) and (2.30) in Equation (2.28) we obtain

$$\begin{aligned} F_{\text{eng}} &= \frac{\eta P_{\text{eng}}(N_{\text{eng}})}{\dot{x}} = \frac{2\pi\eta T(N_{\text{eng}})}{60\dot{x}} \\ &= \frac{2\eta T(N_{\text{eng}}) i_d i_g}{d_{\text{wheel}}} = \frac{\eta T(N_{\text{eng}}) i_d i_g}{r_{\text{wheel}}} \qquad [\text{N}]. \end{aligned} \tag{2.31}$$

For internal combustion engines we need to consider a minimum speed $N_{\text{eng}_{\text{min}}}$: Below this speed the engine is shut down, and Equations (2.28) and (2.31) become undefined or 0. For the sake of simplicity we do not consider clutch dynamics, so to start from standstill we assume the engine to be running at least at $N_{\text{eng}_{\text{min}}}$ rpm. To account for this in Equation (2.28) we compute vehicle's speed in first gear ($i_g = i_{g_1}$) at minimum engine speed by using Equation (2.29):

$$\dot{x}_{\text{min}} = \frac{d_{\text{wheel}} \pi}{60 i_d i_{g_1} N_{\text{eng}_{\text{min}}}}. \tag{2.32}$$

Then, when computing engine power for a given speed we use

$$F_{\text{eng}}(\dot{x}) = \begin{cases} \frac{\eta P_{\text{eng}}(\dot{x}_{\text{min}})}{\dot{x}_{\text{min}}} & \text{if} \quad \dot{x} \le \dot{x}_{\text{min}} \\ \frac{\eta P_{\text{eng}}(\dot{x})}{\dot{x}} & \text{if} \quad \dot{x} > \dot{x}_{\text{min}} \end{cases} \tag{2.33}$$

where $P_{\text{eng}}(\dot{x})$ simply means the power for the engine speed corresponding to vehicle speed $\dot{x}$ and currently engaged gear.

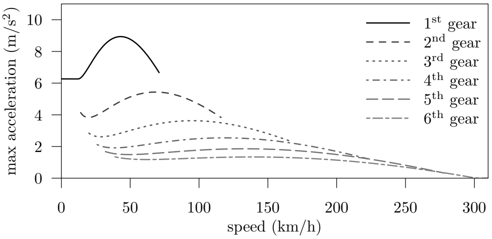


Figure 2.8 – Maximum acceleration curves as a function of vehicle speed and parameterized with respect to the different gears in the real case of an Audi R8 4.2 FSI Quattro car (2007).

For the torque formulation, instead:

$$F_{\text{eng}}(N_{\text{eng}}) = \begin{cases} \frac{\eta T(N_{\text{eng}_{\text{min}}}) i_d i_g}{r_{\text{wheel}}} & \text{if} \quad N_{\text{eng}} \leq N_{\text{eng}_{\text{min}}} \\ \frac{\eta T(N_{\text{eng}}) i_d i_g}{r_{\text{wheel}}} & \text{if} \quad N_{\text{eng}} > N_{\text{eng}_{\text{min}}}. \end{cases} \tag{2.34}$$

Figure 2.8 shows the maximum acceleration curves computed using the model in Equation (2.33) as a function of vehicle speed for all gear ratios using the parameters for an Audi R8 4.2 FSI Quattro car (2007). According to the model, the system can provide any desired acceleration values u, provided that such value is below the maximum acceleration curve. For powerful engines we also need to consider tire slip. Potentially, the engine is capable of providing a really high traction force, but above a certain limit the tires will start to slip limiting the acceleration. The maximum amount of tractive force depends on the normal load on the wheels, i.e.,

$$F_{\text{wheelmax}} = \mu F_{\perp}, \tag{2.35}$$

where μ is the friction coefficient between the tires and the road. The normal force depend on weight, aerodynamic load, weight distribution, etc. In the model, we disregard aerodynamics and changes in the weight distribution due to acceleration and braking maneuvers. Equation (2.35) thus becomes

$$F_{\text{wheelmax}} = \mu m g \gamma, \tag{2.36}$$

where γ is a coefficient that accounts for the type of wheel drive. For example, $\gamma = 1$ for an all-wheel drive, while $\gamma = \frac{1}{2}$ for a front-wheel or a rear-wheel drive. Equation (2.36) needs to be added as further constraint when computing vehicle's maximum acceleration.

Due to the mechanics involved in the engine, the system is affected by an actuation lag τ_{eng} (see Equations (2.25) and (2.27)). The lag depends on fuel injection time τ_{inj}, combustion time τ_{burn}, and transport delay τ_{exh}, i.e., the time needed for exhaust gases to reach the pre-catalyst UEGO-sensor. We can estimate this delay with [68]:

$$\tau_{\text{eng}}(n) = \tau_{\text{inj}}(n) + \tau_{\text{burn}}(n) + \tau_{\text{exh}}, \tag{2.37}$$

where

$$\tau_{\text{inj}}(n) = \frac{2(N_C - 1)}{n \cdot N_C}, \quad \tau_{\text{burn}}(n) = \frac{3}{2n}. \tag{2.38}$$

In Equation (2.38), N_C is the number of cylinders, and n is the engine speed expressed in [rps]. For simplicity, the transport delay τ_{exh} is approximated to a mean value of 100 ms [68, Section 4.1.3].

2.3.2 Brakes Deceleration and Lag

To accurately derive the wheel braking force we need a description of the physical components of the system, including brake disks and pads geometry, friction effects, and detailed physics-based equations for modeling hydraulic pressure inside the master cylinder. In the simulation model we disregard these concepts, and assume the system to have enough force to lock braking wheels. Further, we assume an Antilock Braking System (ABS) which optimizes braking performance, and we neglect aerodynamic forces and non uniform weight distribution on wheels. Similarly to Equation (2.35), we can estimate the maximum braking force as [26]:

$$F_{\text{brake}\,\text{max}} = \mu m g. \tag{2.39}$$

The actual braking force is finally computed as

$$F_{\text{brake}\,\text{max}} = \min(\lambda m u, \mu m g). \tag{2.40}$$

For what concerns the actuation lag τ_{brakes} in Equation (2.27), we fix it to a constant value of 200 ms [37].

2.4 Review of Existing Platooning Simulation Tools

As for any other vehicular application, one of the most important steps is evaluation. In the case of platooning, the evaluation might focus on showing the benefits for the traffic flow [119], on showing the characteristics of a controller [73], on investigating maneuvers and possible interaction with human-driven vehicles [41, 93], or performing large-scale analysis to understand network behavior [94, 95]. The required information can be obtained by means of theoretical [82], experimental [77, 84, 88], or simulative analysis. We can also find studies where the evaluation is carried out using a combination of such methodologies. In general, simulations of platooning systems provide a good tradeoff between level of realism, easiness of implementation, and scale of the scenario. Sometimes, however, simulators lack some features like communication details, vehicle's physics, or availability to the public. By reviewing the literature on simulation platforms for platooning we found that every simulator was lacking a feature because the focus of the study was on a particular aspect.

As an example, Fernandes and Nunes [35] developed a tool based on extensions to the Simulation of Urban MObility (SUMO) [70], implementing the PATH's CACC (Equation (2.16)) as a car following model. The tool, however, assumes all vehicles but the leader to be CACC driven, thus we cannot investigate the impact of legacy vehicles. The road is a single-lane highway, so it is not possible to consider the formation/disruption of platoons, as well as any other maneuver. This is because the authors focus their work on the analysis of the system from a vehicle dynamics perspective, and simply assume a synchronized slotted communication protocol where no interference, collisions, and packet losses occur. Moreover, it is not publicly available.

In [43] the goal is to develop Hestia, a simulator that focuses mostly on physics of the vehicles to test different distributed agent-based models. The simulator features a 3D environment for the maximum realism: It includes object detection through sensors, road conditions, and engine modeling. As for [35], given that the focus is on dynamics, the simulator idealizes the communication part, and does not properly simulate the network at the packet and signal levels. Moreover, the authors do not discuss its scaling properties, and the simulator does not support mixed scenarios.

To study the impact of CACC systems on traffic flow, van Arem et al. [119] develop a stochastic simulation model, which features a multi-lane highway where vehicles have the possibility to overtake each other by changing lane. It is possible to configure and use different vehicle types, as well as to consider both human and automated driving. Again, the networking part is neglected because of the aim of the work, so packet losses and interferences are not considered.

The work by Lei et al. [73] determines the impact of packet loss on the string-stability of a CACC. The simulator developed for the purpose is a complex system where control laws are implemented in Matlab/SIMULINK, road network and vehicles are handled by SUMO, and the network is simulated by OMNeT++. The simulator is, however, tight to the analysis of the specific control system and it is not publicly available Another advanced simulator from the point of view of realistic vehicle dynamics is the one by Zhao et al. [129]. It extends the commercial simulator VISSIM, featuring human-behavioral, ACC, and CACC models. Moreover, it includes some platooning management maneuver, enabling the possibility of studying mixed scenarios with platoon formations and disruptions. The communication part, however, is not implemented, and the simulator is not available to the community.

Other studies care more about the realism of the network, rather than the mobility. Indeed in [19] the authors use a simulator with a full IEEE 802.11p stack as well as detailed fading and shadowing phenomena. Jia et al. [59] extend the Veins framework [107], which provides a fully fledged DSRC/WAVE stack [28]. Their work, however, considers no CACCs but only the IDM car following model. Moreover, the source code is not publicly available.

We ourselves started working on the concept of platooning simulation using the vehicular networking simulator Veins [98]. Yet, this earlier work did not consider the integration of all the needed controller and maneuver models.

By reviewing the literature we have also noticed that every study develops its own simulator, either because the existing ones miss a particular feature or, most probably, because it is not possible to retrieve them online. This led us to develop the Platooning Extension for Veins (PLEXE), which we describe in Chapter 3.

Chapter 3

Simulation Tool: PLEXE

In Section 2.4 we reviewed the literature about platooning simulators, highlighting the features that they miss. Hence, we identified the fundamental characteristics of a platooning simulator, which are:

- Openness: The simulator must be free to download online, and it should be possible to modify it according to specific purposes;
- Active maintenance: It must be kept up-to-date and it must improve with time. The idea is thus to start from a well known and widely used simulator, which will provide a solid base. Moreover, thanks to the openness policy, the community will be able to identify (and fix) potential bugs;
- Realistic networking and dynamics: As cooperative systems heavily rely on communication, the realism of network simulation is crucial. Vehicle dynamics, on the other hand, is also very important, because it permits to understand how a vehicle would behave in particular real world conditions;
- Extensibility: Given the number of Cooperative Adaptive Cruise Control (CACC) algorithms proposed in the literature, a researcher should be able to easily implement new controllers in the simulator. Moreover, it should be also possible to create new traffic scenarios;
- Mixed traffic: The introduction of platooning and CACCs in the real world will be gradual. One important issue is thus the interaction with human-driven vehicles. Hence, the simulator should have the possibility to simultaneously consider both autonomously driven vehicles, as well as human behavioral models.

These are all features of the tool we develop: PLEXE. The name stands for Platooning Extension for Veins: As previously mentioned, indeed, one problem we have found is that each platooning study develops its own simulator. Given the amount of vehicular networking simulators available (Veins, NCTUns, iTetris, TraNS just to name a few [69,83, 107,122]), the solution for sure is not to develop a new one. For this reason we decided to extend Veins, because it is well known, actively maintained, and already features realistic simulation of vehicular communication [28,106,108]. In this chapter we describe the main features of PLEXE, what we implemented, and some use cases. The main references for this chapter are:

- M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *6th IEEE Vehicular Networking Conference (VNC 2014)*. Paderborn, Germany: IEEE, December 2014, pp. 53–60. My contribution in this work was the design and the implementation of the PLEXE simulator.

- M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. Lo Cigno, "Supporting Platooning Maneuvers through IVC: An Initial Protocol Analysis for the Join Maneuver," in *11th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2014)*. Obergurgl, Austria: IEEE, April 2014, pp. 130–137. My contribution in this work was the discovery of potential issues while performing maneuvers in mixed scenario, and the design of the network protocol.

For the sake of completeness we start by describing Veins [107]. Veins is a framework that provides a complete vehicular communication stack, as well as tailored channel models, to the OMNeT++ network simulator [120]. Moreover, it produces realistic node mobility by coupling OMNeT++ with the road traffic simulator called Simulation of Urban MObility (SUMO) [70]. To this aim, for each vehicle traveling in SUMO, Veins instantiates a network node in OMNeT++. Each node in the network simulator is associated with an IEEE 802.11p and IEEE 1609 network stack, a beaconing protocol, and one or more applications running on top of it.

At each SUMO simulation time step, i.e., when vehicles are moved, Veins mirrors the movement in the corresponding OMNeT++ nodes by updating the mobility model. The communication between network and traffic simulator is performed through the TraCI interface exposed by SUMO. Via TraCI, Veins can query SUMO about current simulation status (e.g., number of vehicles, positions, speeds, etc.), or change traffic dynamics by, for instance, re-routing vehicles.

PLEXE, in turn, further extends TraCI interactions to fetch vehicle's data from SUMO that can be sent to other cars, or used by platooning protocols and applications. When

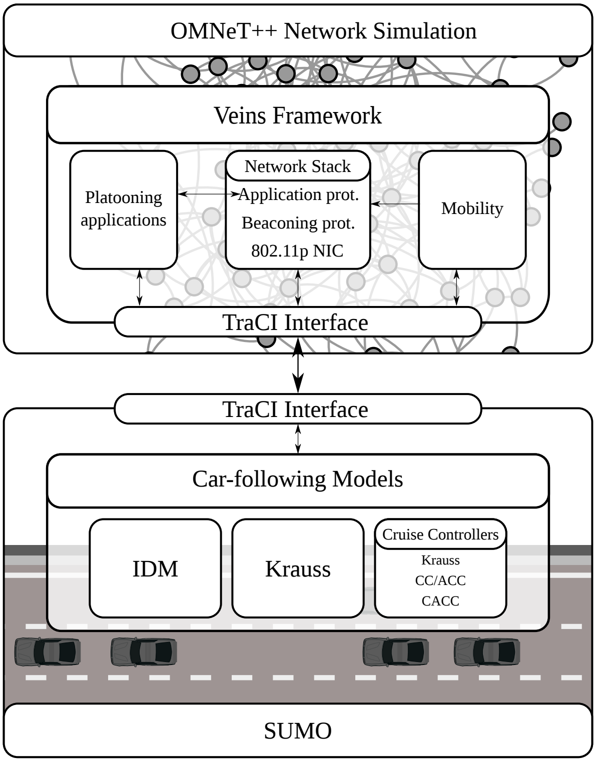


Figure 3.1 – Schematic structure of the simulator. Copyright © 2014 IEEE.

a vehicle (in fact, an OMNeT++ node) receives such data, this data can be sent to the CACCs in SUMO. Communication protocols for platooning and application layer logics are realized in OMNeT++, while controllers and engine dynamics are implemented in SUMO. Figure 3.1 depicts the schematic overview of the extended simulation framework.

To correctly develop platooning models we need to work in two directions: *i)* implement platooning capabilities (i.e., controllers), engine dynamics, and elementary maneuvers in SUMO, and *ii)* protocols and application logic in OMNeT++/Veins. We also require minor changes to both simulators to enhance the bidirectional coupling.

The version of PLEXE we presented in [99] was based on Veins 2.2 and SUMO 0.17.1. In the meantime, we kept PLEXE up to date and, at the time of writing, it was based on Veins 4.0 and SUMO 0.22.0.

3.1 Implementing Platooning Capabilities in SUMO

The implementation introduces a new car-following model enabling both longitudinal acceleration control (using cruise controllers) and a simplified transversal (i.e., steering) control to be able to change lanes. With this new SUMO car-following model, longitudinal controllers described in Section 2.2 are accessible via TraCI.

In general, standard SUMO car-following models are designed to mimic human drivers. Consider as an example the Intelligent Driver Model (IDM) [117], or the Krauss [71] model. The new one developed in PLEXE is called CC, which stands for Cruise Controllers. The idea is to have all Cruise Control (CC), Adaptive Cruise Control (ACC), and CACC models within a single file, and that the user can modify it to implement new ones. More details about coding, examples, and a tutorial can be found in the online documentation available on the official website (`http://plexe.car2x.org`).

By default, CC uses the Krauss model to drive the car. This is useful to simulate realistic scenarios where drivers manually drive the car into a freeway using an on-ramp. The driver would then join a platoon and give control to the system until the desired exit, when he/she would take back control and leave the freeway. The new car-following model includes standard CC/ACC, and three different CACCs [84, 88, 90]. Then, to reproduce engine dynamics, PLEXE provides a generic engine class, which can be extended to implement various models. PLEXE comes with an implementation of a first order lag (Section 2.2.1) and of the realistic model described in Section 2.3.

The TraCI interface permits to access, modify the behavior, and retrieve information from the control and the engine models. In this work we do not list all the functionalities accessible via TraCI, but we just describe the most important ones. Please refer to the online documentation for more details. For what CCs are concerned, the user can change all the important parameters such as desired speed $\dot{x}_d$, time headway T, desired gap d_d, feed the controller with information about any vehicle in the platoon, etc. At runtime it is possible to choose the active controller, which can either be the human behavioral model, the CC, the ACC, or any of the CACCs provided by PLEXE or implemented by the user.

We can also override the standard controller behavior to perform testing or to verify the safety of a system. One example is the possibility to set a fixed acceleration to the leader to simulate an emergency braking. Moreover, we can retrieve distance to route end, or set a fixed lane a vehicle should stay in.

The parameters for the engine models depend on the model itself. The first order lag only requires the sampling time Δ_t and the time constant τ, while the parameters for the realistic engine model are specified in an XML file, as they are several. Listing 3.1 shows a

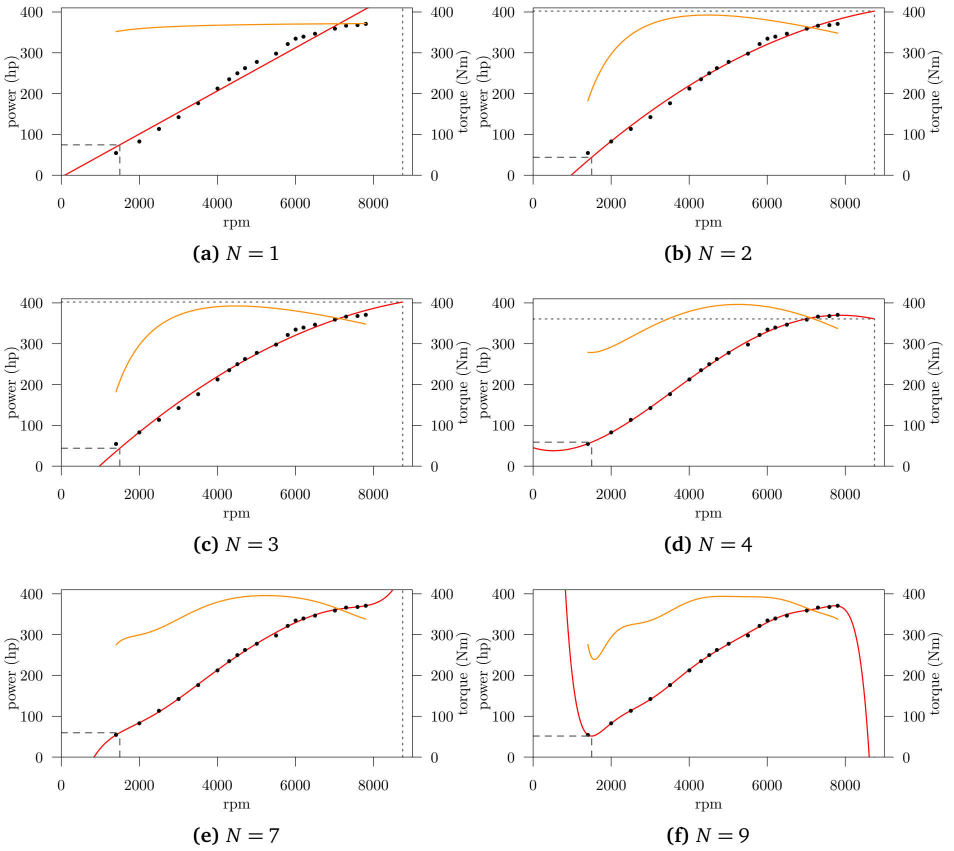
```xml
<vehicles>
  <vehicle id="audi-r8" description="Audi R8">
    <!-- Gearbox ratios -->
    <gears>
      <gear n="1" ratio="4.373"/>
      <gear n="2" ratio="2.709"/>
      <gear n="3" ratio="1.878"/>
      <gear n="4" ratio="1.411"/>
      <gear n="5" ratio="1.126"/>
      <gear n="6" ratio="0.928"/>
      <differential ratio="3.462"/>
    </gears>
    <!-- Mass in kilograms and mass factor considering engine ↩
        rotational inertia -->
    <mass mass="1628" massFactor="1.089"/>
    <!-- Tracting wheels diameter in meters, friction coefficient, ↩
        and parameters for rolling resistance -->
    <wheels diameter="0.66" friction="1" cr1="0.0136" cr2="5.18e-7"/>
    <!-- Air friction coefficients. Drag coefficient and maximum ↩
        section in squared meters -->
    <drag cAir="0.30" section="2.1"/>
    <!-- Engine characteristics including power mapping parameters -->
    <engine efficiency="0.9" cylinders="8" type="poly" ↩
        maxRpm="8750" minRpm="1500" tauEx="0.1">
      <power x0="45.1859637473846" x1="-0.0291861008036701" ↩
          x2="3.07914798702012e-05" x3="-3.74768401980442e-09" ↩
          x4="1.23531013014598e-13"/>
    </engine>
    <!-- Gear shifting rules -->
    <shifting rpm="8500" deltaRpm="200"/>
    <!-- Brakes data. Tau is the actuation time in seconds. Other ↩
        parameters such as friction are taken from the wheels ↩
        section -->
    <brakes tau="0.2"/>
  </vehicle>
</vehicles>
```

Listing 3.1 – Sample engine configuration.

sample XML configuration file including a single vehicle, but there can be more than one. Most parameters refer to the ones of the model described in Section 2.3. The `id` is used to choose the desired vehicle type via TraCI. Engine power characteristics are simulated using a polynomial of the form

$$P_{\text{eng}}(N_{\text{eng}}) = \sum_{i=0}^{N} x_i N_{\text{eng}}^i, \tag{3.1}$$

where N_{eng} is the engine speed in [rpm], x_i are the coefficients of the polynomial specified in the XML file, and N is automatically computed depending on the number of coefficients. Equation (3.1) is the polynomial that better fits the real power curve of the engine. The polynomial must be manually chosen by the user, but PLEXE provides an R script that takes in input the measurements points, minimum and maximum rpm, and plots the fitting for



(a) $N = 1$

(b) $N = 2$

(c) $N = 3$

(d) $N = 4$

(e) $N = 7$

(f) $N = 9$

Figure 3.2 – Polynomials of different degrees fitting measured power curve. The black dots represent the measured points, the red curve the fitting polynomial, and the orange line is the corresponding torque curve.

degrees going from $N = 1$ to $N = 9$. The user can then compare the polynomials and choose the most appropriate one, copy-pasting the XML code printed by the script directly into the configuration file. The script cannot automatically choose a particular polynomial, as higher degrees have lower residuals but they might overfit. Figure 3.2 shows different polynomial fits of the same measurement points. The linear ($N = 1$) and the highest degree ($N = 9$) are too unrealistic, but for $N = 4$ the fit is good, reproducing the typical loss of power around the maximum number of rpm as well.

The remaining configuration parameters, which are not part of the model described in Section 2.3, are the ones to perform gear shifting. In the XML, we need to specify a desired engine speed N_{shift} at which we want to perform upshifting plus an amount Δ_{shift} to avoid oscillations around the shifting point. More formally, let $\{i_{g_i}\},\ i = 1,\dots,N_{\text{gears}}$ be the set of available gear ratios with $i_{g_i} > i_{g_{i+1}},\ i = 1,\dots,N_{\text{gears}} - 1$. When the car accelerates ($\ddot{x} >= 0$) we can use Equation (2.29) to choose the minimum gear i which satisfies

$$\frac{60 i_d i_{g_i} \dot{x}}{d_{\text{wheel}} \pi} < N_{\text{shift}} + \Delta_{\text{shift}}, \tag{3.2}$$

where $\dot{x}$ is the current vehicle speed. Similarly, when the car decelerates ($\ddot{x} < 0$) we choose the minimum gear i such that

$$\frac{60 i_d i_{g_i} \dot{x}}{d_{\text{wheel}} \pi} < N_{\text{shift}} - \Delta_{\text{shift}}. \tag{3.3}$$

3.2 Platooning Protocols and Applications in Veins

As described in the previous section, the majority of the changes regard SUMO. In Veins, besides the required changes to the TraCI interface, PLEXE provides a network stack comprising an IEEE 802.11p network interface card, a basic message dissemination protocol, and an application layer on top of the message distribution. The aim of the protocol layer is implementing the communication strategy for sharing data among platooning vehicles. PLEXE provides a base protocol class that implements basic functionalities that inheriting classes can use. This includes logging of statistics, send and receive primitives, and parameters loading. The idea is to have subclasses focus on beaconing strategy only.

The same principle is used for the base class at the application layer: It loads simulation parameters, or it passes data to the CACC via TraCI. Another task the application layer is in charge of is the management of cars and platoons. For example, it is a duty of the application to decide whether a particular car is the leader of a platoon, in which lane it should travel, if it has to join or leave the platoon, etc. It is thus simple to describe and implement platooning applications thanks to the primitives provided by the highest layer. The online documentation details example primitives and applications beyond standard car-following. In this thesis, we do not go into the coding details but only describe the idea and show some results.

The protocol plus application layer stack is provided as a sample structure, but it can be modified or substituted depending on the needs. Still, PLEXE provides all TraCI commands that are fundamental to communicate with the SUMO car-following model.

As previously mentioned, one feature provided by the simulator is the possibility of having both automated and human-driven vehicles. It is indeed possible to setup standard SUMO traffic flows of human vehicles, which will interfere with platooning cars both at the mobility and at the network level.

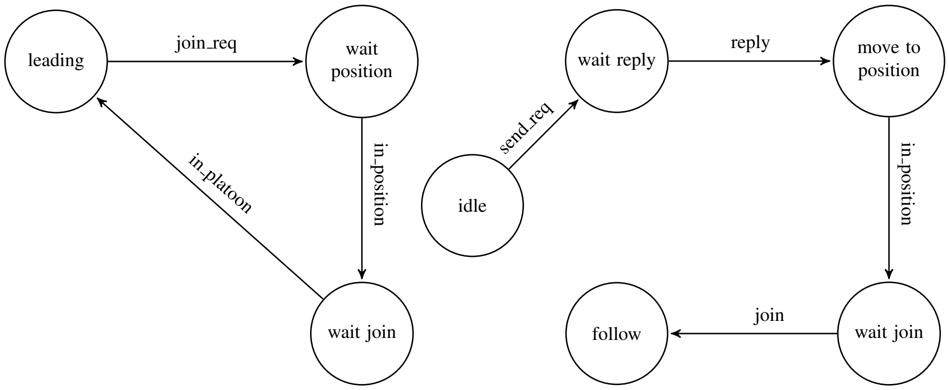
3.3 Platoon Maneuvering

The implementation and testing of a controller for platooning is one among several issue we want to investigate. The CACC simply maintains the platoon, but this platoon needs to be created, modified, and disrupted. Moreover, we need to cope with interfering cars, or we might want to overtake a slower vehicle or platoon. Basically, we need to think to a platoon as a single "flexible" vehicle driven by an intelligent (or autonomous) driver. Studying, developing, and testing maneuvers is a fundamental part of platooning research [44, 72, 76, 93], and PLEXE is a suitable framework for the purpose.

To support these features we choose to provide protocol primitives such as join request/response, together with basic building blocks using the primitives to implement simple maneuvers. Here we provide a simple example for the sake of clarity. The documentation provides more details about the implementation, while [93] performs the analysis of more complex maneuvers.

In particular, we consider a join-at-tail maneuver with two main actors: The leader and the joiner. Each actor has its own state machine (Figure 3.3) that drives the protocol coordinating the maneuver. The other followers, which are already part of the platoon, have no active role, but they are informed about the ongoing maneuver by overhearing the exchange of messages. This way, the system management can, for example, inform the drivers about the additional vehicle.

Figure 3.3 shows that joining and leading vehicles start from the IDLE and LEADING states respectively. The joiner uses the `send_req` primitive ("join platoon" is a parameter of the primitive) to request the leader to join: Then, it moves to the WAIT REPLY state. The leader positively answers using the `join_req` primitive including information such as lane, join position, etc., and moves to the WAIT POSITION state. The joiner approaches the platoon and, once in the position negotiated during the request, it notifies the leader about its ability to join. The leader confirms and the joiner enables the CACC, closing the gap to its predecessor. Finally, the leader switches back to the LEADING state and the joiner to FOLLOW. The entire maneuver can be implemented by simply using the additional TraCI commands provided by PLEXE, and requires no further changes to SUMO.



(a) leader **(b)** joiner

Figure 3.3 – State machines of the sample join maneuver. Copyright © 2014 IEEE.

Table 3.1 – Network and road traffic simulation parameters. Copyright © 2014 IEEE.

	Parameter	Value
communication	Path loss model	Free space ($\alpha = 2.0$)
	PHY model	IEEE 802.11p
	MAC model	1609.4 single channel (CCH)
	Frequency	5.89 GHz
	Bitrate	6 Mbit/s (QPSK R = $^1/_2$)
	Access category	AC_VI
	MSDU size	200 B
	Transmit power	20 dBm
mobility	Leader's average speed	100 km/h
	Oscillation frequency	0.2 Hz
	Oscillation amplitude	$\simeq$ 95 km/h to 105 km/h
	Platoon size	8 cars
	Car length	4 m
controllers	Engine lag τ	0.5 s
	Weighting factor C_1	0.5
	Controller bandwidth ω_n	0.2 Hz
	Damping factor ξ	1
	Desired gap d_d	5 m
	Headway time T	0.3 s and 1.2 s
	ACC parameter λ	0.1
	Distance gain k_d	0.7
	Speed gain k_s	1.0
	Desired speed $\dot{x}_d$ (followers)	130 km/h

3.4 Sample Use Cases

This section provides two use cases to demonstrate the versatility of Plexe. The first use case is for comparing the performance of a new, fictional controller against the ACC and CACC algorithms provided by the simulator. The second one, instead, implements the join-at-back maneuver described in Section 3.3: In particular a car joins a 4-car platoon traveling on the same freeway. In addition, we report the results of our study on a join-at-middle maneuver [93]. Finally, we present a simple example to demonstrate how the realistic engine model performs. In here we show the results obtained with a single-shot run, but simulations can obviously be repeated by the OMNeT++ framework to obtain statistical confidence.

3.4.1 Controller Analysis

Implementing a new controller in PLEXE is straightforward and well documented in the online documentation. For this reason we do not describe all the required implementation steps but we only provide the control formula for the fictional controller, which we will call TESTCC:

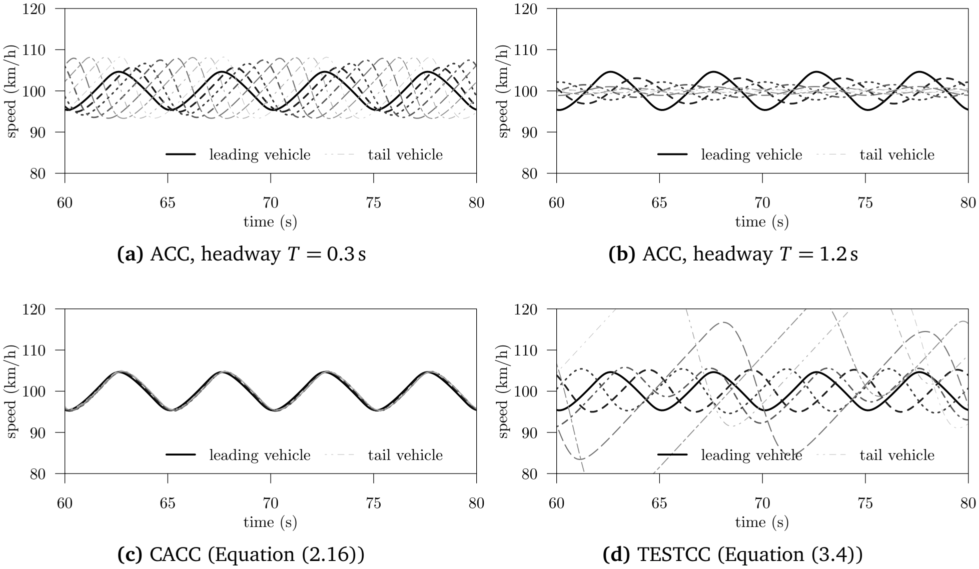
$$u_i = k_d\,(x_{i-1} - x_i - l_{i-1} - 25\,\mathrm{m}) + k_s\,(\dot{x}_{i-1} - \dot{x}_i)\,. \tag{3.4}$$

The goal of the controller is to maintain the same speed of the vehicle in front and a fixed inter-vehicle gap of 25 m. The distance and the speed error terms have two design gains, i.e., k_d and k_s, respectively. We assume the distance to be provided by the radar, while the speed to be obtained through Vehicle-to-Vehicle (V2V) communication.

TESTCC is tested against the ACC and the PATH CACC (Equations (2.13) and (2.16)) in a platoon of eight cars traveling on a freeway, with the leader continuously changing its speed in a sinusoidal fashion. Table 3.1 summarizes simulation's parameters.

We compare the controllers by looking at the speed profiles as function of time (Figure 3.4). In the plot, darker and thicker lines represents vehicles at the front of the platoon, while the thinner and lighter the ones at the back.

We first compare the ACCs using different headway times. When violating the string stability constraint (Equation (2.14)), i.e., when setting $T = 0.3\,\mathrm{s}$, the followers amplify the leader-induced disturbance (Figure 3.4a). As a result, the speed amplitude increases towards the end of the platoon. Conversely, for $T = 1.2\,\mathrm{s}$ (Figure 3.4b), the vehicles are able to progressively attenuate the disturbance. This, however, results in a tracking lag, as shown by the phasing between vehicles' speed profiles. The CACC does not cause this behavior (Figure 3.4c), as each car perfectly tracks the acceleration and the speed inputs broadcast by the leader. Figure 3.4d shows the unstable behavior of TESTCC. Vehicles at the tail of the platoon amplify the disturbance in an uncontrolled manner: While the leader's speed oscillates between 95 km/h and 105 km/h, some vehicles at the tail have a speed amplitude exceeding 80 km/h and 120 km/h.



(a) ACC, headway $T = 0.3\,\text{s}$

(b) ACC, headway $T = 1.2\,\text{s}$

(c) CACC (Equation (2.16))

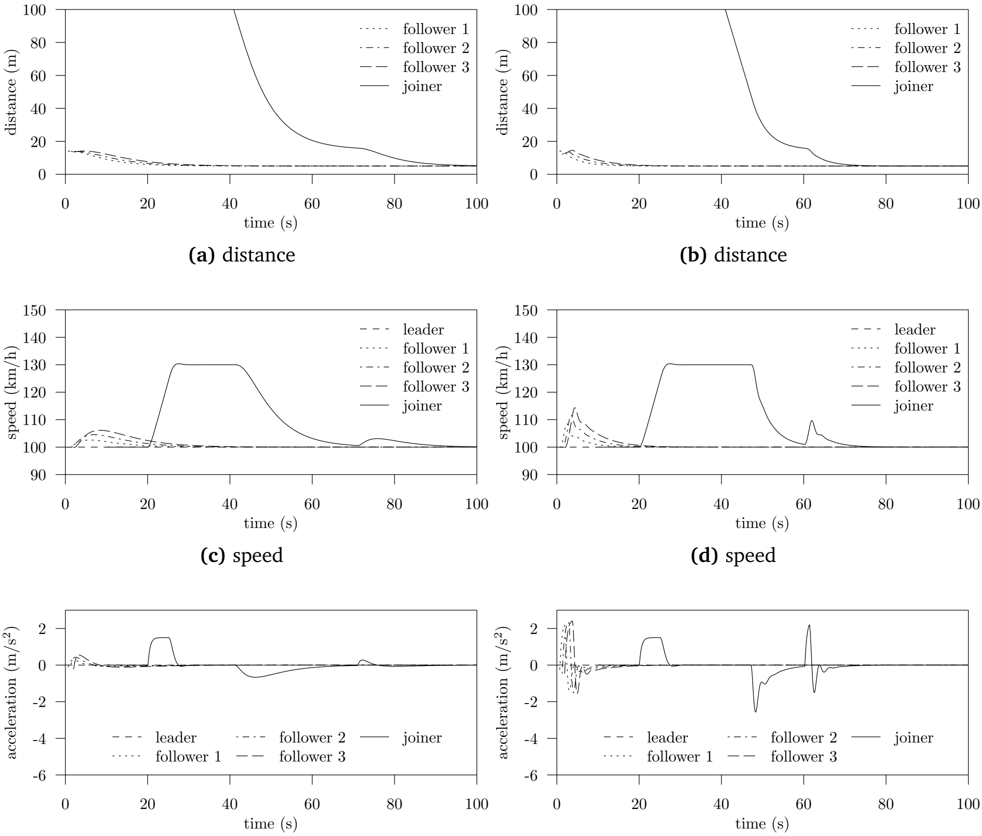
(d) TESTCC (Equation (3.4))

Figure 3.4 – Vehicles speed for the different implemented cruise controllers showing string-stability properties. Copyright © 2014 IEEE.

3.4.2 Join Maneuver

To perform the analysis of the sample join maneuver we employ the same scenario parameters of Table 3.1, but the leader drivers at constant speed, the platoon is made by 4 cars only, and, in addition, we test the performance of the CACC for additional values of the control parameters, i.e., $\xi = 2$ and $\omega_n = 1\,\text{Hz}$. Figure 3.5 shows the results of the simulations: In particular, we plot distance, speed, and acceleration as a function of time for the two CACC settings. The first 20 s are needed to reach the steady state: The three followers join the simulation and close the gap to the leader. The joiner vehicle is the last entering the simulation with a speed of 100 km/h. This vehicle requires the leader to join and, after a positive answer, it accelerates up to 130 km/h and reaches the platoon. Once within 15 m from the last vehicle, the joiner requests a confirmation to conclude the maneuver and closes the gap (5 m).

It is interesting to observe the change in dynamics caused by different parameterization of the controller. When $\xi = 2$ and $\omega_n = 1\,\text{Hz}$, the controller converges faster and concludes the maneuver at around 70 s, roughly half a minute earlier than the other setup. The fast convergence setup, however, leads to undesired oscillations and might result in uncomfortable accelerations for the passengers. This shows that PLEXE can also be used to



(a) distance **(b)** distance

(c) speed **(d)** speed

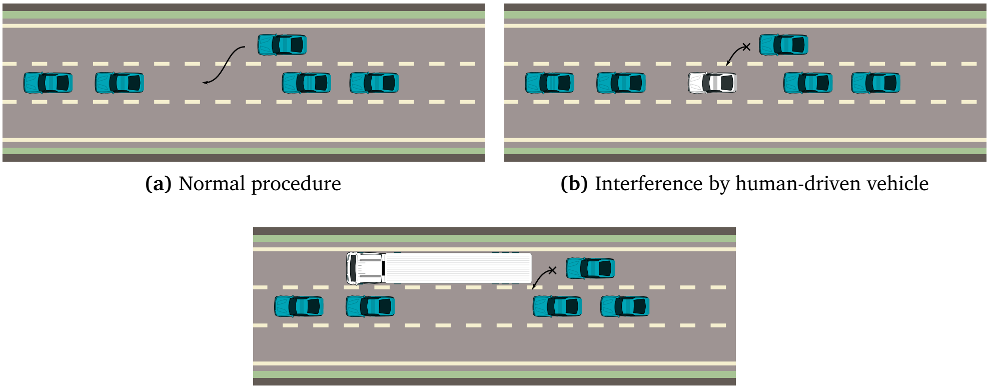
(e) acceleration **(f)** acceleration

Figure 3.5 – Vehicle dynamics for the join maneuver. The left column shows the results for $\xi = 1$, $\omega_n = 0.2\,\text{Hz}$, while the right column for $\xi = 2$, $\omega_n = 1\,\text{Hz}$. Copyright © 2014 IEEE.

tune the parameters of a controller to meet a good tradeoff. In this specific case, a good tradeoff between convergence time and driving comfort.

3.4.3 Human-driven Vehicles Interference

The previous use case considers a simple join-at-back maneuver on a completely dedicated highway. In reality, the road will be shared between human-driven and automated vehicles, and we need to consider potential interferences caused by the formers when designing protocols for maneuver management. In [93], we investigate the issues connected to a join-at-middle maneuver. Figure 3.6a shows a successful maneuver: The joining vehicle



(a) Normal procedure

(b) Interference by human-driven vehicle

(c) Interference by slower vehicle in front

Figure 3.6 – Graphical sketch of different situations for the join-at-middle maneuver. Automated cars shown in dark color. Copyright © 2014 IEEE.

moves into the position indicated by the leader and enters into the gap left by the followers. More in detail, we have three vehicles "actively" involved in the procedure: L, F, and M. The leader L coordinates the maneuver, while the follower F creates the gap to let the joiner M enter the platoon. The following list summarizes the procedure:

1. M discovers the platoon led by L by listening to its beacon and sends a request for joining;

2. L, decides whether to deny or accept the request depending on some policies. In case of a deny, L sends a negative response and concludes the maneuver. Otherwise, L positively replies and indicates in which position M should join (i.e., in front of F). The leader also includes the identities of the vehicle F and the one in front of F;

3. M approaches the platoon using the information sent by the vehicle in front of F. In real life this step could be done by the human driver, which would receive indications from the interface. In this case we assume that the procedure is automated, thus data received from the vehicle in front of F is fed into the CACC which brings M in the correct position automatically;

4. Once in position, M informs L, which, in turn, commands F to open the gap;

5. F executes the maneuver and informs L;

6. L communicates M to change lane;

7. M changes lane, closes the gap and informs L;

8. L sends to F a message for closing the gap;

9. F closes the gap and informs the leader when done.

During the maneuver, however, some events can prevent its conclusion. As an example, a human driven vehicle can enter the gap reserved to the joiner (Figure 3.6b), or the joiner might be prevented from reaching the platoon due to a slower vehicle ahead (Figure 3.6c).

PLEXE can be used to develop and test a protocol capable of detecting and reacting to these situations. Our work in [93] is an example. There we develop a protocol, which implements the aforementioned steps and takes care of detecting and reacting to dangerous situations. For example, to cope with slower vehicles ahead, the system continuously checks the control input u of the ACC, even when the system is switched off. If the value of u becomes smaller than $-3\,\mathrm{m/s^2}$, the system decides that the car is coming too close to the vehicle in front and aborts the maneuver. To handle the interference by human-driven vehicles, instead, the car that opens the gap continuously cross checks the distance to the front vehicle measured by the radar to the distance computed via GPS. If the system detects a discrepancy between the two values for a certain amount of time it informs the protocol, which aborts the maneuver.

We test the protocol in four scenarios. The basic scenario reproduces a single car that wants to join a four-vehicle platoon in the middle. Then, we consider four different variations of such a scenario:

- Scenario 0 (no interference): This reproduces the case in Figure 3.6a, where no vehicle interferes with the maneuver;

- Scenario 1 (far truck interference): In this case the joiner approaches a truck (Figure 3.6c), but the truck is far enough for the maneuver to complete;

- Scenario 2 (close truck interference): This is the same as Scenario 1, but in this case the track is closer to the joiner, so the maneuver needs to be aborted;

- Scenario 3 (car interference): A human driven car enters the gap reserved to the joining vehicle. The interfering vehicle leaves after a few seconds but the maneuver is aborted for safety reasons, leaving the initial platoon split into two 2-car platoons.

Figures 3.7 and 3.8 show the dynamics of the vehicles in the platoon, plus the dynamics of the joiner, for the four considered scenarios. In particular, Figures 3.7a, 3.7c, 3.8a and 3.8c show the distance of each vehicle from the leader in time. Figures 3.7b, 3.7d, 3.8b and 3.8d show instead the distance as perceived by the radar, i.e., to the vehicle directly in front. If the radar detects no vehicle, no line is drawn on the plot.

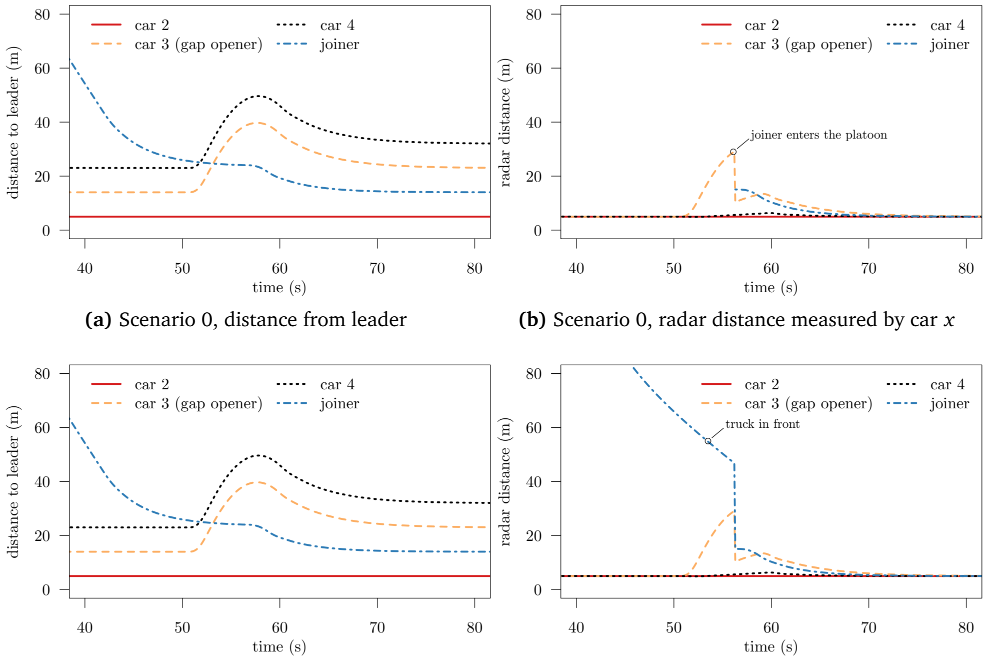
To better understand the graphs, we start by analyzing the maneuver for Scenario 0 (Figures 3.7a and 3.7b). At the beginning of the simulation, cars 2, 3, and 4 are traveling in a platoon following their leader. In Figure 3.7a, this is shown by the three horizontal lines around 40 s indicating the positions of the vehicles with respect to the leader. Figure 3.7b, instead, plots three overlapping lines indicating that each vehicle has a 5 m distance to the car in front, as measured by the radar. In Figure 3.7a we then see the joiner coming close to the platoon, i.e., decreasing its position relative to the leader. The radar plot does not initially shows a line for the joiner because the latter is traveling on a different lane, so no vehicle ahead is detected. When the joiner is close enough to the platoon, car 3 decelerates to open the gap, and so does car 4 to stay behind car 3. Figure 3.7b shows that the gap between cars 3 and 2 increases up to roughly 25 m: At that point, the joiner enters the platoon by changing lane. Finally, the maneuver concludes with the joiner and car 3 closing the gap.

Figures 3.7c and 3.7d show the dynamics of Scenario 1, i.e., the one where the joiner approaches a slower truck ahead, which is far enough not to disturb the maneuver. The plots are very similar to the previous ones, with the only difference that the radar plot for the joiner this time shows the distance from the truck up until roughly 57 s (50 m distant), where the joining car changes lane to enter the platoon and successfully conclude the maneuver.

In Figures 3.8a and 3.8b (Scenario 2), instead, while car 3 is opening the gap, the joining vehicle comes too close to the truck and the system decides to abort the maneuver for safety reasons. Car 3 thus closes the gap, leaving the platoon as before starting the maneuver, while the joiner leaves the control to the ACC and remains behind the truck at a safe distance.

The final two plots (Figures 3.8c and 3.8d) show the dynamics for Scenario 3. While opening the gap (at around 53 s) a human driven vehicles enters such gap. The radar of car 3 detects the danger and, even if the human driven vehicle leaves after a few second, it decides to abort the maneuver. Car 3 thus form a two-vehicle platoon with car 4 and decelerates to keep a safe distance from car 2.

These results show the flexibility of PLEXE, which can be used to simulate scenarios with both automated and human-driven vehicles, to consider dangerous situations, and to implement and evaluate protocols and maneuvers for platooning.



(a) Scenario 0, distance from leader

(b) Scenario 0, radar distance measured by car *x*

(c) Scenario 1, distance from leader

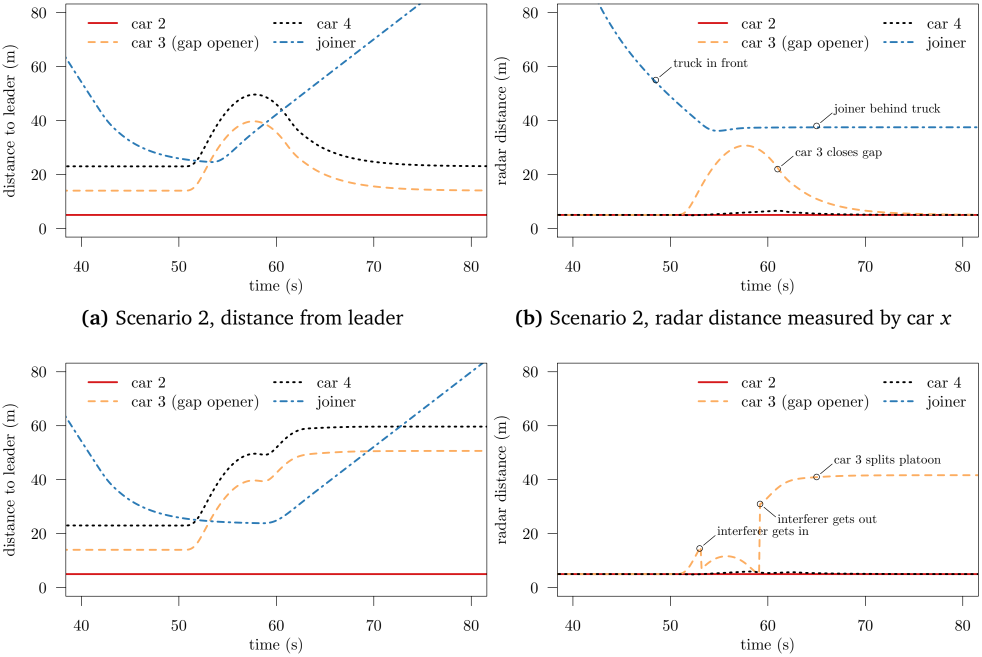
(d) Scenario 1, radar distance measured by car *x*

Figure 3.7 – Vehicles dynamics for scenarios 0 and 1: left plots show the GPS distance to the leader; right plots show the measured radar distance. Copyright © 2014 IEEE.

3.4.4 Engine Model

As a final demonstration we take a simple example showing how the realistic engine model performs. In particular we configure three vehicles with the characteristics of an Alfa Romeo 147, an Audi R8, and a Bugatti Veyron. We do not list all the vehicle parameters here, but they can be found online, together with the source code of the example, on the official webpage. In the simulation, we let the vehicles start from standstill and require the maximum possible acceleration from the engine. Vehicles continuously accelerate for 60 s and then brake down to a complete stop using the maximum possible deceleration.

Figure 3.9 shows accelerations and speed profiles for the three vehicles. The acceleration shows the difference between the three engines. Moreover, it is possible to see how wheel force drops when gear switching occurs, and how the maximum acceleration approaches 0 due to air resistance: This effect naturally reproduces how the cars reach their maximum speeds, as shown in Figure 3.9b. During the braking phase it is also possible to see the effect of the air drag: When vehicles starts to brake they decelerate



(a) Scenario 2, distance from leader

(b) Scenario 2, radar distance measured by car *x*

(c) Scenario 3, distance from leader

(d) Scenario 3, radar distance measured by car *x*

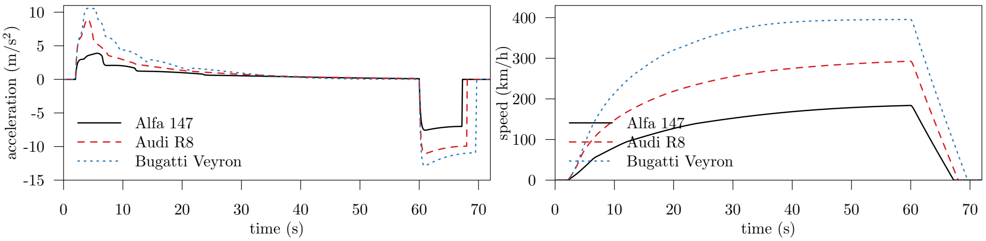
Figure 3.8 – Vehicles dynamics for scenarios 2 and 3: left plots show the GPS distance to the leader; right plots show the measured radar distance. Copyright © 2014 IEEE.

stronger because air is helping them to slow down. Air resistance then diminishes with speed, and the car only resorts to its own braking system. Finally, the simulator accounts for the different type of tires mounted, showing better braking capabilities for sporty cars.

This examples shows how PLEXE can be used to study control algorithm performance in the presence of inhomogeneous vehicles, such as trucks, commercial, or luxury cars.

3.5 Conclusion

In this section we described PLEXE, a framework for the realistic analysis of a platooning system. PLEXE provides several longitudinal controller, including standard CC/ACC, as well as modern, state of the art CACCs. The key features of this framework are the easiness of implementing and customizing new controllers, the ability of performing mixed traffic scenarios, and the realistic simulation of both wireless networking and vehicle dynamics. Furthermore, PLEXE is free to download from the official website. In this section we have



(a) acceleration

(b) speed

Figure 3.9 – Vehicles acceleration and speed profiles when using the realistic engine model.

seen two sample use cases (controller analysis and maneuver implementation), but the framework is not limited to that and can be used to investigate many different scenarios and platooning issues. All these features make PLEXE a valid research tool before the real world deployment of platooning systems.

Chapter 4

Safe and Efficient Communication for Platooning

In this chapter we describe the work we made towards a communication protocol for the platooning application. In fact, in this thesis we focus on Cooperative Adaptive Cruise Control (CACC)-related communication, as platooning encompasses more than that. Indeed data sharing is not only needed to control the vehicles, but also to manage the platoons and perform maneuvers. The aim of our work is understanding to which extent the plain IEEE 802.11p network stack is able to support the CACC and determine the requirements of the latter. Upon this, we incrementally develop a protocol that is safe and at the same time efficiently uses network resources, even in heavily dense scenarios. In the first phase we identify how to support the CACC in a static (i.e., using periodic beaconing) configuration. In the second one, we release this constraint and try to improve the efficiency by dynamically adjusting the beaconing period depending on vehicle dynamics.

The work in this chapter is based on the following publications:

- C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, July 2015. In this work we present Dynamic Beaconing (DynB), the dynamic beaconing protocol described in Section 2.1.2. DynB, together with ETSI Decentralized Congestion Control (DCC), is taken as a reference protocol for comparison. My contribution was the set up a subset of the simulations and the analysis of the outcome.

- M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards Inter-Vehicle Communication Strategies for Platooning Support," in *7th IFIP/IEEE International Workshop on Communication Technologies for Vehicles (Nets4Cars 2014-Fall)*. Saint-Petersburg, Russia: IEEE, October 2014, pp. 1–6. In this work we propose our static beaconing approaches and perform an initial comparison with the reference protocols[2]. My contribution for this work was conceiving the different protocols.

- M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards Communication Strategies for Platooning: Simulative and Experimental Evaluation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5411–5423, December 2015. In this work we extend the analysis of the previous one by considering the full DCC protocol specification, by analyzing the impact of the Clear Channel Assessment (CCA)-threshold, by investigating the coexistence of our approaches with DCC in a mixed scenario, and by studying the impact of communication on CACC performance. My contribution was the implementation of a more sophisticated physical layer for Veins and the full DCC protocol.

- M. Segata, F. Dressler, and R. Lo Cigno, "Jerk Beaconing: A Dynamic Approach to Platooning," in *7th IEEE Vehicular Networking Conference (VNC 2015)*. Kyoto, Japan: IEEE, December 2015, pp. 135–142. In this work we develop a dynamic beaconing protocol for platooning. My contribution was the design of the beaconing algorithm and of the lower-layer reliability protocol.

4.1 Application Layer Requirements

We propose a set of protocols that work on top of the IEEE 802.11p/IEEE 1609.4 PHY/MAC: All messages scheduled by those protocols contend for the channel in a CSMA/CA fashion. We remind the reader that our focus in on the CACC developed in the PATH project [86,88] (Equation (2.16)). The input to the system obtained through wireless communication are thus leader's and front vehicle's speed and acceleration. For designing the algorithms we exploit such properties, i.e., we assume that each vehicle knows its position within the platoon and uses this information to decide *how* and *when* to send a beacon.

To decide *how* we need to consider that only the leader needs to reach all vehicles in the platoon. The followers only need to share their speed and acceleration with the

[2] In this preliminary work we consider only the Transmit Rate Control (TRC) component of the ETSI DCC protocol.

vehicle immediately behind. We can thus reduce the transmit power to favor the spatial reuse of the channel and avoid to interfere with cars that are “not interested” in receiving such data. The leaders can instead use a higher power to reach all platoon’s vehicles.

Dynamic transmit power adjustment is in general very challenging as it needs to cope with highly dynamic networks [113]. For the application at stake, however, the design is simplified by the topology of the platoon, but different type of vehicles might be an issue. A truck with a front antenna, for example, might not be able to communicate with a car immediately behind [15], requiring an ad-hoc power calibration to compensate for shadowing. In this thesis we disregard vehicle shadowing effects and assume a fixed transmit power of 0 dBm for all followers.

To decide *when* to send, we exploit the position of a vehicle inside the platoon. In particular, we let the leader send as first, then the second, the third, and so on in a cascading fashion. This is not a standard Time Division Multiple Access (TDMA) approach, because in TDMA every node obeys the same rules. In this approach TDMA is implemented only between vehicles belonging to the same platoon to reduce intra-platoon channel contention.

Algorithm 4.1 shows the code for this pseudo-slotted approach. We refer to the slotted protocols as SLB and SLBP (without and with transmit power control, respectively). We divide leader’s inter-beacon interval into slots and each follower uses the slot corresponding to its position. The leader initiates the procedure by sending the first beacon at protocol startup, which is used by its followers for synchronization. Depending on their position and a time offset (i.e., the slot time) they compute when to schedule the next beacon.

```
function onStartup
    if myRole == leader then
        schedule(sendBeacon, beaconInterval)
function sendBeacon
    sendBroadcast(getVehicleData())
    schedule(sendBeacon, beaconInterval)
function onBeacon(beacon)
    updateCACC(beacon)
    if beacon.sender == leader then
        onLeaderBeacon(beacon)
function onLeaderBeacon(beacon)
    unschedule(sendBeacon)
    schedule(sendBeacon, myPosition · offset)
```

Algorithm 4.1 – SLB protocol.

Moreover, upon sending a beacon, each follower schedules another send event for the next beacon interval to prevent a missing leader beacon from locking the protocol. This "backup" event is deleted upon successful reception of a leader's beacon.

The idea behind the protocol is to synchronize the nodes within a platoon and reduce random channel contention. As a side effect, even without specific inter-platoon cooperation, the leaders can end up synchronizing with other platoons when performing CSMA/CA at the MAC layer.

We consider a platoon size of 20 cars and we thus divide the beacon interval in 20 slots of 5 ms each. The leader always broadcasts data frames using a transmit power of 20 dBm. The followers use the same amount of power when transmit power control is disabled: Otherwise they use 0 dBm.

We compare SLB and SLBP with a baseline approach, i.e., with static, periodic broadcasting using only CSMA/CA as channel access control and the same transmit power control of SLBP. We refer to the baseline protocols as STB and STBP.

4.2 Experimental Validation

To validate and calibrate the network models we employ in the simulations we performed some measurements with real cars (Figure 4.1). We drove four cars on a private road at 20 km/h to safely maintain an inter-vehicle distance of 5 m. Vehicles were driven by humans in respect of the Austrian legislation. The system was autonomously recording experiment data without required input from the driver.

To perform the measurement campaign we used two Cohda Wireless MK2[3] and two Unex DCMA-86P[4] devices, both IEEE 802.11p compliant. We employed four (one per device) Mobile Mark ECOM9-500 dipole antennas with a gain of 9 dBi which were magnetically mounted on the rooftops.

We implemented STB and SLB as applications sitting on top of the MAC layer and performed repeated experiments on a 2 km stretch of road. Each experiment had a duration of roughly 30 s and was repeated three times to change the environmental conditions. We configured the radios and the protocols using the parameters in Table 4.1. To compare the results and to calibrate the simulation model, we reproduced the same conditions (number of cars, protocols, parameters, etc.) in a simulated scenario.

[3] `http://www.cohdawireless.com/`

[4] `http://www.unex.com.tw/`

Table 4.1 – Parameters employed in the experimental validation. Copyright © 2015 IEEE.

Parameter	Value
Beacon frequency	10 Hz, 20 Hz and 25 Hz
Tx power (leader)	20 dBm
Tx power (followers)	20 dBm, 10 dBm and 0 dBm
Modulation	QPSK R=½

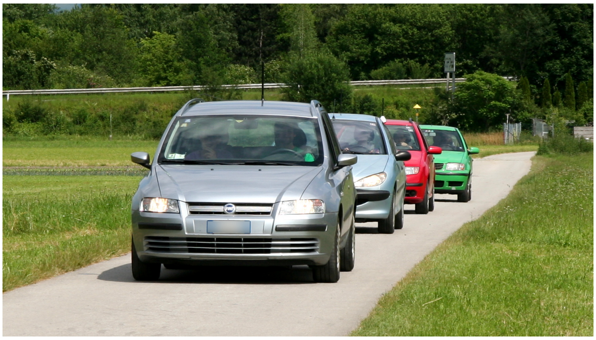


Figure 4.1 – Cars used for the experimental validation. Copyright © 2014 IEEE.

As a comparison metric we use the distribution of the received power. In such an experiment, indeed, packet reception rate was always above 99 %, making it unusable for comparison.

Given the setup, we can model fading at the received using a Rice distribution with a strong Line Of Sight (LOS) component. This is reasonable because in this work we only consider cars. Boban et al. [18], indeed, show that if the first Fresnel zone is obstructed by less than 40 %, then shadowing causes no major impact on signal strength. We verified this statement in another measurement campaign [96] where we tested the impact of different vehicles as obstacles. When considering a strong LOS component with Rician fading, we can approximate the amplitude of the received signal by a Lognormal distribution [8].

To test the equipment before analyzing the data we paired the Network Interface Cards (NICs) through a cable and a 90 dB attenuator, and recorded received signal powers. One device was incorrectly calibrated and was using a transmit power lower than the configured one and was reporting incorrect received signal strength. For this reason we equalized received power values to compensate device errors.

Figure 4.2 compares the simulation and the experimental results for a leader transmit power of 20 dBm and followers transmit power of 0 dBm. Regarding the shape of the distribution, we see that different experiments show slightly different standard deviations.

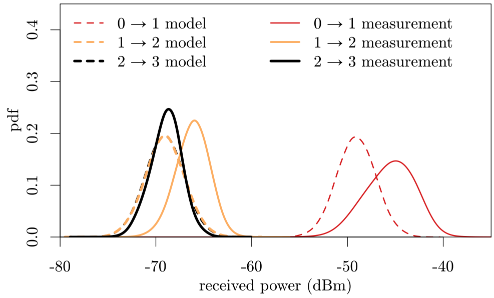


Figure 4.2 – Comparison of received power distributions between the experimental testbed (measurement) and the simulation environment (model), for the communication between immediate followers (0 being the leader, 3 being the last car). Transmit power 20 dBm for the leader and 0 dBm for the followers. 1 → 2 and 2 → 3 model curves overlap because the distance and the transmit power between vehicles 1-2 and 2-3 are the same. Copyright © 2015 IEEE.

A good compromise between all of them is to set $\sigma = 2\,\text{dBm}$ which matches the LOS measurements in [96] as well. For what concerns the average received power, instead, we can observe that it is higher for the real world experiments. This is caused by the high gains of the antennas we used in the experiments. The simulations, configured with a Free space path loss with an exponent $\alpha = 2.0$, consider a theoretical isotropic antenna with no gain, which is a common assumption for vehicular networking simulations. In this work we are mainly interested in the shape of the distributions rather than the exact quantities to ensure better comparability with other studies.

4.3 Analysis of STB and SLB

To compare the different approaches we use the PLEXE framework described in Chapter 3. The aim is to understand the characteristics and the behavior of the protocols and network conditions in a “stressful” configuration. The analysis includes STB, STBP, SLB, SLBP, as well as two dynamic approaches described in Section 2.1.1, i.e., DynB and ETSI DCC. We first analyze all protocols on a global scale, both from a networking and an application layer perspective. Then, we focus on showing the delay requirements of the application by testing the performance of the CACC in an emergency braking scenario.

To configure the simulations we used the parameters listed in Table 4.2. For what concerns channel models, we employ a Free space propagation loss with an exponent $\alpha = 2.0$ coupled with a log-normally distributed fading with a standard deviation $\sigma = 2.0$, as for Section 4.2. We used PHY and MAC layer models provided by Veins and presented

Table 4.2 – Network and road traffic simulation parameters. Copyright © 2015 IEEE.

	Parameter	Value
communication	Path loss model	Free space ($\alpha = 2.0$)
	Fading model	Log-normal ($\sigma = 2.0$)
	PHY/MAC model	IEEE 802.11p/1609.4 single channel
	Frequency	5.89 GHz (CCH)
	Bitrate	6 Mbit/s (QPSK R = ½)
	Access category	AC_VI
	MSDU size	200 B
	Transmit power	20 dBm and 0 dBm
	Sensitivity	−95 dBm
	Noise floor	−95 dBm
	CCA-threshold	−95 dBm, −85 dBm and −65 dBm
mobility	Number of cars	160, 320 and 640
	Number of lanes	4
	Platoon size	20 cars
	Car length	4 m
	Intra-platoon distance	5 m
	Inter-platoon distance	$\simeq$ 41 m
	Speed	100 km/h
controllers	C_1	0.5
	ω_n	0.2 Hz
	ξ	1
	T	1.5 s
	λ	0.1
	τ	0.5 s
DynB	I_{des}	0.1 s
	b_{des}	0.25

in [29]. The IEEE 1609.4 switching mechanism between CCH and Service Channel (SCH) has been disabled, and we only used the CCH. We used a physical layer bitrate of 6 Mbit/s, the optimal one for vehicular safety applications [60].

The application layer generates packets with an MSDU size of 200 B associated with the AC_VI access category. Messages are generated ten times a second (10 Hz), as a minimum assumed requirement for CACC [84]: DynB and DCC generate messages using their own scheduling policies. Furthermore, DynB uses a static transmit power of 20 dBm. DynB implementation and configuration parameters are taken from [110], while DCC parameters are set to their default values given in the standard (Table 2.1). All protocols send standard, unacknowledged broadcast Cooperative Awareness Messages (CAMs), as defined in the ETSI standard [34].

Finally, for all protocols but DCC, we test different CCA-thresholds. The CCA-threshold indicates the amount of energy required by the physical layer to declare the channel busy when missing the preamble portion of a frame. This might happen, for example, because a node was transmitting a frame, or when multiple frames are simultaneously received (cumulative interference). We test CCA-threshold values of −95 dBm (equal to model's minimum sensitivity), −85 dBm (the minimum required sensitivity for the lower modulation and coding scheme), and −65 dBm (as defined by the standard [3, 18.3.10.6]. Notice that, for the IEEE 802.11 standard, sensitivity and CCA-threshold define two different parameters. The sensitivity is the amount of power above which 90 % of the preambles are correctly detected [3, 18.3.10.6]. In our model, which is a common assumption in physical layer simulations, the sensitivity is the amount of energy needed to detect an incoming frame: Frames below such amount of power are simply ignored. We set this value to −95 dBm.

Concerning mobility and dynamics, we simulate a 4-lane freeway with platoons of 20 cars each, for a total number of 160, 320, and 640 cars. The choice of the number of cars might seem unreasonable, but we use such a high number of vehicles for two reasons. First, we want to know if the protocols stop working above a certain vehicle density, meaning that we are interested in finding an upper limit, if any. Second, large freeways might easily get congested and reach such densities during rush hours, and we want to guarantee that platooning works even in these conditions: Understanding whether it can be supported or not is crucial. With 640 cars the interference domain does not cover the entire scenario, i.e., cars at the front of the considered stretch of road do not hear cars at the back and vice versa. Thus adding further cars does not affect the ones in the middle, and results with more cars do not change anymore.

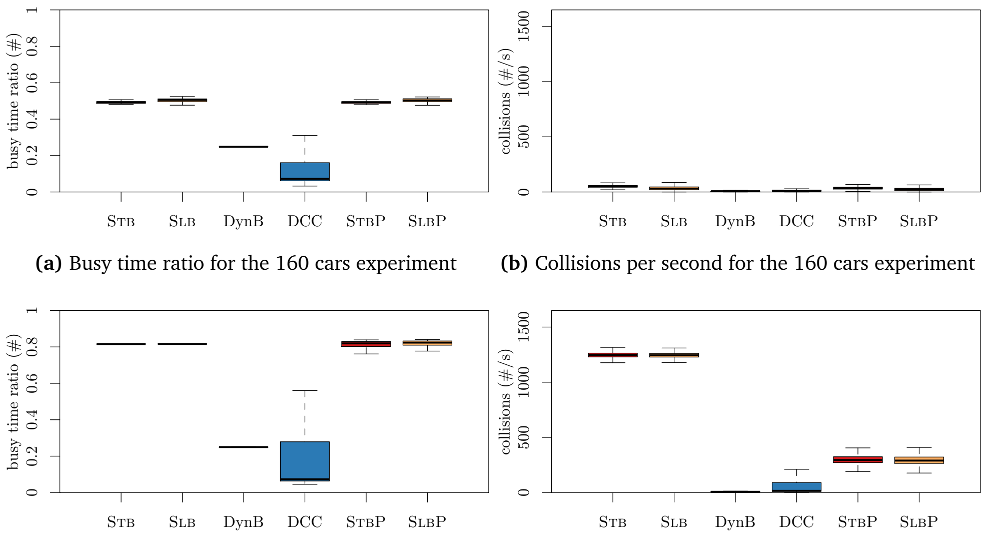
Finally, other platooning relevant parameters are the CACC desired distance d_d for the followers, set to 5 m, and the platoon speed, set to 100 km/h. The cruising speed is maintained by the leaders, which are controlled by a standard Adaptive Cruise Control (ACC) with a time headway $T = 1.5$ s. We maintain a constant speed to focus on network analysis only. Vehicle dynamics are considered in Sections 4.3.5 and 4.4.

We repeat each combination of density and protocol 10 times to obtain statistical confidence for the results. Moreover, we partially removed border data to avoid biasing the evaluation with border effects. We give more details on the data processing procedure when discussing each metric.

4.3.1 Generic Network Performance

The analysis in Sections 4.3.1 to 4.3.3 assume a dedicated platooning channel to ease the interpretation of the results and obtain fundamental understanding of the behavior of the network. We relax this assumption in Section 4.3.4 and mix automated vehicles running our protocol with human driven vehicles running DCC.

We start the analysis considering two generic network metrics: Channel busy ratio and collisions. Channel busy ratio is the fraction of time the channel is declared busy by the



(a) Busy time ratio for the 160 cars experiment

(b) Collisions per second for the 160 cars experiment

(c) Busy time ratio for the 640 cars experiment

(d) Collisions per second for the 640 cars experiment

Figure 4.3 – Busy time ratio and collisions for the 160 and the 640 cars scenarios, CCA-threshold set to −95 dBm. Copyright © 2015 IEEE.

physical layer in certain time frame. The busy ratio is sampled and recorded once a second by all vehicles in the simulation. The collisions, instead, measure the number of frames not decoded by a vehicle due to interferences, and it is recorded once a second as well. We group sampled values into boxplots displaying the first and third quartiles as a box, the median as a center line, and absolute minimum and maximum using whiskers. Before grouping the data, we discard values collected during simulation warm-up, ignoring the transient phase. Further, for the 640 cars scenario, we remove the data of 7.5 % of the vehicles at the head and at the tail respectively to account for border effects. Vehicles at borders, indeed, experience a lower channel congestion because their interference domain is smaller than scenario's size and thus falsify the results by showing a higher performance than the actual one. We thus pre-analyzed the data to verify that removing 15 % of the cars was enough to get rid of this effect.

We consider busy ratio and collisions for the 160 and 640 cars experiments and a CCA-threshold of −95 dBm (Figure 4.3). A number of vehicles up to 160 does not cause network overload. In the worst case the channel load is around 50 %, and the amount of collisions is limited. DynB is capable of maintaining the channel load at the desired level (25 %) while, due to its dynamic behavior, DCC makes the channel load span between 5 %

and 20 %. For STB and SLB, transmit control in such a low density scenario is not helpful, because the network is not saturated. Moreover, the performance improvement of the slotted approaches with respect to SLB and SLBP is minor, and statistically irrelevant.

For the 640 cars scenario, the results change significantly. The dynamic approaches adapt to the high density of nodes and they are capable of maintaining the network load and the collisions under control. Both static and slotted approaches (without transmit power control), instead, result in a channel load around 80 % and a large amount of collisions, thus completely saturating the channel. In this particular case, using transmit power control in STBP and SLBP is beneficial, as even if the channel load is close to complete saturation, the amount of collisions lowers dramatically with respect to STB and SLB.

4.3.2 Application Layer Perspective

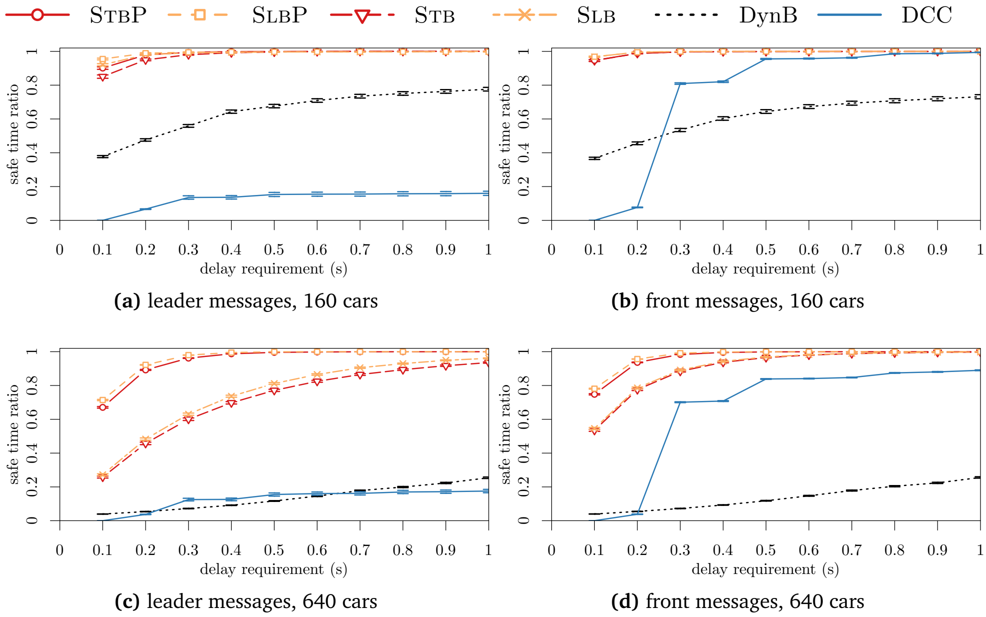
The analysis in Section 4.3.1 revealed the effectiveness of the dynamic approaches in keeping network utilization under control, outperforming all static and slotted approaches. DCC and DynB, however, are designed to improve the overall network conditions and do not, in general, consider specific application requirements. For a platooning application, missed or omitted packets can be harmful, thus posing passengers' safety at risk. A CACC without a backup mechanism needs to perform a "blind" control action when missing data packets, i.e., it simply assumes the input variables to be unchanged. In the worst case, this can result in instabilities or crashes. At this stage of the work, however, to understand the effectiveness of the protocols from the application layer perspective we would need precise theoretical requirements of the controller.

To overcome this issue we define an application layer metric, which can be configured and computed for a maximum tolerable delay δ_{req}. We define with $\mathcal{D}$ the set of all the inter-message delays collected by a vehicle. The subset of $\mathcal{D}$ satisfying the delay requirement is defined as

$$\mathcal{D}_{\text{safe}} = \left\{ d : d \in \mathcal{D} \wedge d \leq \delta_{\text{req}} + \Delta \right\}, \tag{4.1}$$

with Δ being a small grace period that accounts for uncertainties like MAC layer backoffs. During this grace period, however, the information is still useful for the controller. In particular, we choose $\Delta = 10\,\text{ms}$, i.e., the CACC controller sampling time [84,99]. We can now define a metric that measures the amount of time a vehicle is in a safe state, and we call it safe time ratio metric r_{safe}:

$$r_{\text{safe}} = \frac{\sum_{d_s \in \mathcal{D}_{\text{safe}}} d_s}{\sum_{d \in \mathcal{D}} d}. \tag{4.2}$$



(a) leader messages, 160 cars

(b) front messages, 160 cars

(c) leader messages, 640 cars

(d) front messages, 640 cars

Figure 4.4 – Safe time ratios of both leader and front messages for the 160 and the 640 cars scenarios, CCA-threshold set to −95 dBm. Copyright © 2015 IEEE.

To understand this metric, imagine all delays in $\mathcal{D}$ to be 200 ms and that the delay requirement δ_{req} is 100 ms. In this particular case, $r_{\text{safe}} = 0$, meaning that the vehicle was never in a safe state. If the requirement is instead $\delta_{\text{req}} = 300$ ms, $r_{\text{safe}} = 1$, meaning that all information was timely received.

Figure 4.4 shows the r_{safe} metric for the protocols we consider. Each point in the plots is the average safe time ratio among all cars and all simulations for a particular configuration with a 95 % confidence interval. As before, we remove data collected by border vehicles in the 640 cars scenario. Furthermore, for the same scenario, we do not consider the inter-arrival time of leader messages for the first platoon of each lane, as for DynB such leaders send beacons at a higher frequency because of their border position. The metric is split between leader and front messages, because each vehicle expects to receive data from platoon's leader and from the vehicle immediately in front.

By changing perspective from the network to the application layer, Figure 4.4 shows the consequences of maintaining a predefined busy ratio. DynB, to reach its goal, needs indeed to increase the beacon interval indefinitely. In the 160 cars scenario the performance is still acceptable, but in the most demanding scenario, even for $\delta_{\text{req}} = 1$ s, the vehicles are safe for less than 30 % of the time. One characteristic of DynB is its fairness: Indeed the safe time ratio is similar for both leader and front messages.

On the contrary, the behavior of DCC is "orthogonal". The performance of DCC does not depend on the number of vehicles, but r_{safe} for front messages is higher than for leader message, thus being unfair. DCC unfairness is caused by the low transmit power used in the RESTRICTIVE state (−10 dBm) which makes communication with the leader hardly possible.

From the upper-layer perspective STBP and SLBP bring enormous benefits because they specifically consider application layer requirements. Even if STB and SLB perform better than the dynamic approaches, their transmit power control counterparts reach way higher safe time ratio in the highest density scenario. For 640 cars and $\delta_{\text{req}} = 100\,\text{ms}$, the use of transmit power control provides a gain in performance of about 40 % and 20 %, for leader and front messages, respectively. In the worst case, i.e., leader messages in the 640 cars scenario, for the most demanding delay requirement (100 ms) STBP and SLBP ensure that vehicles are in a safe state roughly 70 % of the time. The performance for front vehicle messages is clearly higher due to the lower distance between the communicating nodes. Overall, the slotted approach give a slight advantage, but it is not significantly better. Obviously, the benefits provided by the slotted approach depend on the size of the platoon: The larger the platoon, the smaller the contention among vehicles belonging to the same platoon.

The safe time ratio metric shows that, even if STBP and SLBP generate more collisions (Figure 4.3), they concern data frames that the CACC application is not interested in. For what regards leader and front messages, instead, STBP and SLBP deliver them within 200 ms in 90 % of the cases. The dynamic approaches, to avoid a large number of collisions, need to lower the beacon rate, at the expenses of extremely large delays. In conclusion, the results highlight that higher collisions do not necessarily lead to worse application layer performance.

4.3.3 Impact of CCA-threshold

This section briefly analyzes the impact of different CCA-thresholds on SLB and SLBP. As previously mentioned, the CCA-threshold is the amount of energy that the physical layer considers for declaring the channel as busy when missing the preamble portion of the frame. This can occur, for example, when a frame is starting to be received while the radio is in transmit mode. This, when switching back to receive mode, requires the physical layer to measure the amount of energy in the channel to understand whether there is

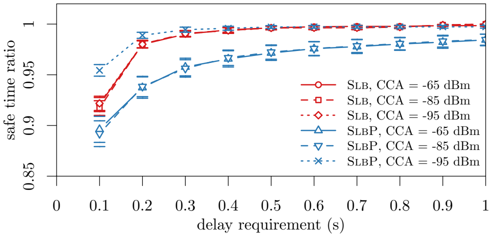


Figure 4.5 – Safe time ratio of leader messages for the slotted approaches, 160 cars scenario, for different values of CCA-threshold. Copyright © 2015 IEEE.

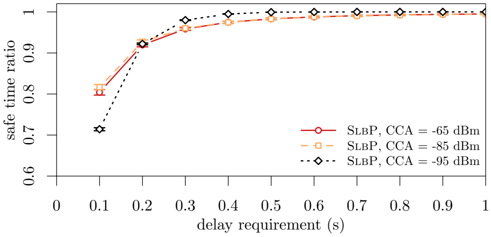


Figure 4.6 – Safe time ratio of leader messages for the slotted approach with transmit power control, 640 cars scenario, for different values of CCA-threshold. Copyright © 2015 IEEE.

ongoing communication or not. The CCA-threshold mandated by the standard is −65 dBm, but we consider −85 dBm and −95 dBm as well.

We compare the safe time ratio metric for the different values of CCA-threshold we consider. In particular, Figure 4.5 shows r_{safe} for the scenario with 160 vehicles and both SLB and SLBP. When using full transmit power in a low density scenario, there is no difference in performance between the CCA-thresholds we consider. On the contrary, when transmit power control is enabled, using a lower CCA-threshold increases the awareness about other vehicles, improving safe time ratio of about 5 % for the lower delay requirements.

A CCA-threshold of −95 dBm, however, is too conservative in high density scenarios, reducing spatial utilization. Figure 4.6 indeed shows that choosing a safe time ratio of −95 dBm worsen the safe time ratio of about 10 % with respect to higher CCA-thresholds for $\delta_{\text{req}} = 0.1$ s.

The analysis suggests that dynamic CCA-threshold adaptation such as in DCC or in [92] would be the best approach. If such a mechanism is unavailable, however, the IEEE 802.11 mandated CCA-threshold of −65 dBm provides, on average, the best performance.

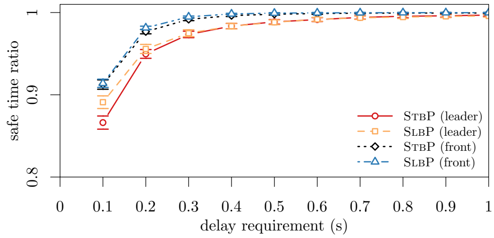


Figure 4.7 – Safe time ratio of leader and front messages for STBP and SLBP, for the scenario with human driven vehicles. CCA-threshold set to −65 dBm. Copyright © 2015 IEEE.

4.3.4 Coexistence with DCC

The analysis in Sections 4.3.1 to 4.3.3 assumed a dedicated communication channel for the platooning application. It is clear, however, that the introduction of automated driving systems will be gradual, thus there will be human-driven vehicles sharing the road and the communication channel as well. To understand the impact of human-driven vehicles we thus modify the scenario with 640 cars and fill two out of four lanes with 320 platooning vehicles running STBP and SLBP, and the remaining two lanes with human driven vehicles using DCC.

Figure 4.7 shows r_{safe} for leader and front messages and both STBP and SLBP. Due to the size of the scenario, we apply the border removal procedure explained in Section 4.3.2. The results in Figure 4.7 are comparable to the ones in Figure 4.4. The safe time ratio is no less than 85 % for a $\delta_{\text{req}} = 100$ ms, for all beaconing approaches. Both STBP and SLBP are unaffected by DCC, because the latter reacts to the network load caused by platooning protocols by switching to the RESTRICTIVE state, which means using a transmit power of −10 dBm. For this reason, DCC suffers a large amount of packet losses as it is not designed to coexist with other protocols: This would require a complete re-parameterization, which, however, is out of the scope of this thesis.

4.3.5 Impact of Communication on CACC Performance

The aim of this section is to understand the requirements of the CACC we considered so far by means of a basic study. In particular we simulate a 20-car platoon using SLB to avoid any kind of channel contention among the vehicles. The platoon cruises at 130 km/h and performs a full-stop braking maneuver with different deceleration rates and beacon frequencies. We then record the minimum inter-vehicle gap among any consecutive pair

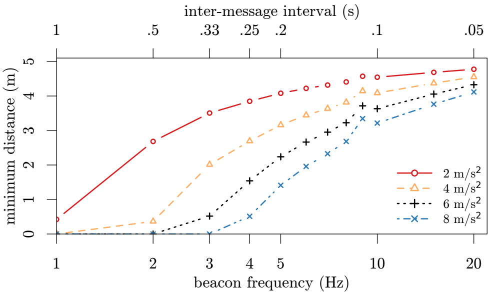


Figure 4.8 – Minimum distances after the complete stop of the platoon as function of the beacon interval, for different leader decelerations. Copyright © 2015 IEEE.

of cars as a measure of system's safety. This permits to give an approximate value to δ_{req}, which was unknown so far.

We setup different scenarios with beacon rates from 1 Hz to 10 Hz in steps of 1 Hz, and from 10 Hz to 20 Hz in steps of 5 Hz. When performing the braking maneuver, the leader decelerates at a rate of 2 m/s^2, 4 m/s^2, 6 m/s^2 and 8 m/s^2. The maximum braking deceleration for follower vehicles is set to 9 m/s^2: The followers thus can brake stronger than their leader. This might not be the case in reality. For example, the real world experiments in [130] try to test a braking scenario using trucks with inhomogeneous maximum decelerations. In our experiments, however, we are more interested in the impairments caused by inter-message delay, rather than by vehicle capabilities.

Each simulation setup is repeated 10 times and, among all repetitions, we take the minimum distance over all pairs of consecutive cars, for each configuration of beacon frequency and deceleration rate.

The results in Figure 4.8 thus show a worst case analysis. The x-axis is in a logarithmic scale to highlight the behavior of the CACC for beacon rates up to 10 Hz. The upper abscissa reports the corresponding inter-message arrival for the sake of comparison with the δ_{req} values used in Figure 4.4. In the plot, a minimum distance of 0 m means that two vehicles crashed into each other.

The graph shows that the delay requirement actually depends on the deceleration rate, i.e., the stronger the deceleration the tighter the requirement. If we consider a 2 m/s^2 deceleration, a beacon inter-arrival of 0.5 s (i.e., a beacon rate of 2 Hz) is enough to avoid a crash, as the worst case minimum distance is around 2.5 m. Conversely, at the maximum deceleration rate (8 m/s^2), an update rate of 3 Hz (0.33 s beacon inter-arrival) might result in a crash. It should be noted that an avoided crash does not necessarily mean feeling safe: When coming too close to the car in front the passengers might feel uncomfortable.

In any case, Figure 4.4 suggests that an inter-arrival larger than 0.3 s can harm system's safety and cause crashes. Notice that this is the case for the particular setup we consider here, i.e., controller parameters, actuation lag, etc. Different parameterizations might have different requirements.

In summary, considering the performance of the protocols in Figure 4.4c and comparing them to the results of the emergency braking scenario (Figure 4.8), we can conclude that STBP and SLBP are safe for the CACC application we consider, as they are able to provide the application with a minimum update rate of 5 Hz in 90 % of the time. This is also due to the robustness of the controller to "blind control" actions. On the other hand, the dynamic protocols would not be able to guarantee system's safety as their update time is larger than 1 s 80 % of the time.

Moreover, Figure 4.8 raises an interesting question: Given that the performance of the controller actually depends on the dynamics of the vehicles, can we develop a dynamic beaconing mechanism able to support a CACC application? The next section answers this question.

4.4 A New Dynamic Approach: Jerk Beaconing

In the previous section we compared two static against two dynamic protocols, showing the benefits of considering application layer requirements in the design of networking protocols. In this section, we make a step forward and we propose a dynamic protocol, following the findings of Section 4.3.5: The requirements in terms of update rate should depend on the dynamics of the vehicles. We can intuitively say that if the measured signals required by the controller do not change, there is no gain in repeatedly transmitting such information. Even if information changes, we can predict missing data, as already witnessed by previous works in the literature [13, 48]. In particular, if a vehicle travels with a constant acceleration $\ddot{x}$, and such a vehicle sent its acceleration $\ddot{x}(t_s)$ and speed $\dot{x}(t_s)$ at time t_s, then all other vehicles that need to know its up to date speed, can estimate it by computing

$$\dot{x}(t) = \dot{x}(t_s) + \ddot{x}(t_s) \cdot (t - t_s) . \tag{4.3}$$

Theoretically speaking, a CACC could be fed with this predicted information with no loss in performance. In reality, we need to take two factors into consideration. The first one is awareness: Completely stopping information sharing is not possible because there is no way of knowing if this "silence" is because the state of a vehicle did not change or because the network failed. We can easily tackle this problem by introducing a minimum

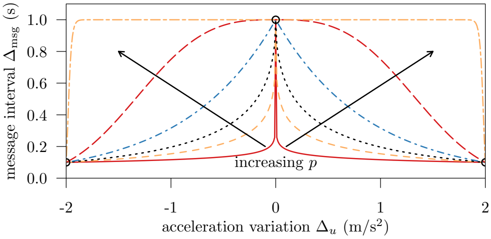


Figure 4.9 – Δ_{msg} function for $\max_{\text{bi}} = 1\,\text{s}$, $\min_{\text{bi}} = 0.1\,\text{s}$, $\Delta_{u_{\max}} = 2\,\text{m/s}^2$, and different values of p (0.1, 0.3, 0.5, 1, 3 and 100). Copyright © 2015 IEEE.

beacon rate as a "heartbeat". The second problem is defining what constant acceleration means, because there exist no such thing in the real world. If we send a beacon for every tiny acceleration variation we would congest the network more than when using static beaconing. There is thus a bound below which changes in acceleration are only noise and can be ignored.

To face this problem we use the concept of jerk, which is a physical quantity measuring the variation of acceleration. Jerk is denoted with $\dddot{x}$ and it is formally defined as

$$\dddot{x}(t) = \frac{d\ddot{x}(t)}{dt}. \tag{4.4}$$

In our approach we exploit a discrete estimation of $\dddot{x}$ to decide whether to send a new beacon or not. In particular, we use the difference between the current CACC control input u (the desired acceleration) and the value sent in the last beacon as a decision variable, i.e.: $\Delta_u = u - u_{\text{sent}}$. To map our decision variable Δ_u to the target beacon interval, we use the following formula:

$$\Delta_{\text{msg}}(\Delta_u) = \max\left(e^{-a|\Delta_u|^p} \cdot b, \min_{\text{bi}}\right). \tag{4.5}$$

In here we use the desired acceleration u instead of the actual one $\ddot{x}$ because u is what we encapsulate in the beacons. In Equation (4.5) we have three tuning parameters a, b, and p. By changing a and b we can choose the maximum ($\max_{\text{bi}}$) and the minimum ($\min_{\text{bi}}$) beacon interval. In particular, we can set a and b as

$$b = \max_{\text{bi}}, \quad a = -\ln\left(\frac{\min_{\text{bi}}}{\max_{\text{bi}}}\right) \cdot \Delta_{u_{\max}}^{-p}. \tag{4.6}$$

to obtain $\Delta_{\text{msg}}(0) = \max_{\text{bi}}$ and $\Delta_{\text{msg}}(\Delta_{u_{\max}}) = \min_{\text{bi}}$. Finally, the parameter p controls protocol's reactivity. In particular, for $p \to 0$, the function returns the minimum beacon interval $\min_{\mathrm{bi}}$ for the slightest change in the desired acceleration. Conversely, for $p \to \infty$, for any Δ_u satisfying $-\Delta_{u_{\max}} < \Delta_u < \Delta_{u_{\max}}$, the beaconing control function returns the maximum beacon interval. Figure 4.9 shows a graphical representation of Equation (4.5) for different values of p.

It is clear that p drives the performance of the protocol. If p is a small quantity, then we should maximize the performance of in terms of safety at the expenses of network overloading. On the contrary, if we increase p we use less network resources, but we might harm passengers' safety. The key is thus to find a value of p which is a good trade off between safety and network efficiency.

4.4.1 Lower Layer Reliability

The *Jerk Beaconing* protocol of Equation (4.5), from a theoretical point of view, changes the sampling frequency of the system in a dynamic way, trying to use the minimum allowable. When using such a mechanism, however, if we loose a frame we violate the minimum sampling rate constraint, which can clearly harm system's safety. *Jerk Beaconing* thus needs to be coupled with a lower layer protocol ensuring message delivery or capable of informing the application the network failed in doing so.

To design this protocol we start by considering the slotted protocol described in Section 4.1 which, besides reducing intra-platoon channel contention, it permits the use of *chain acknowledgements*. Imagine vehicles sending packets using a slotted approach, i.e., first the leader, then the second vehicle, then the third, and so on. Each follower can piggyback an acknowledgement, which can be overheard by its front vehicle directly in the beacon being sent.

Further, we improve the chain acknowledgement mechanism by considering an acknowledgement map. In particular, each vehicle maintains a map $\mathcal{A} = (a_1, \ldots, a_{N-1}) \in \mathbb{N}^{N-1}$, with a_i being the sequence number of the last packet of vehicle $i-1$ acknowledged by vehicle i, and includes $\mathcal{A}$ in each beacon sent. Thanks to the acknowledgement map, platooning members can obtain missing acknowledgements by any vehicle behind them. Moreover, each vehicle re-propagates the latest leader information known, as this is the most important for ensuring the stability of the platoon [86]. More formally, each beacon B_i sent by vehicle i is an n-tuple

$$B_i = (u_i, \dot{x}_i, s_i, \mathcal{A}_i, u_0, \dot{x}_0, s_0), \tag{4.7}$$

where s_i is the sequence number associated to data of vehicle i. The actual data size depends on the number of vehicles in the platoon and the floating point precision we

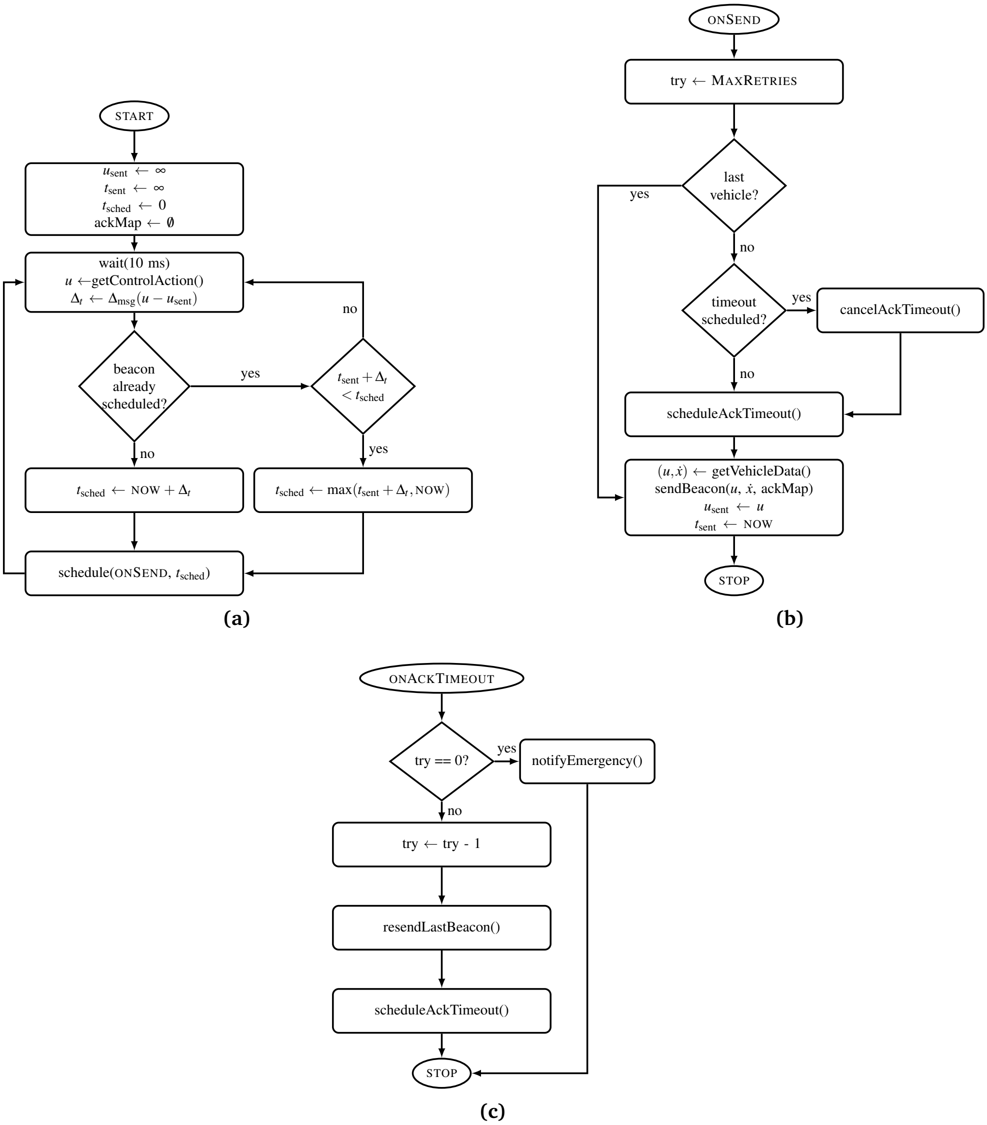

START
$u_{\text{sent}} \leftarrow \infty$
$t_{\text{sent}} \leftarrow \infty$
$t_{\text{sched}} \leftarrow 0$
ackMap ← ∅
wait(10 ms)
$u \leftarrow$getControlAction()
$\Delta_t \leftarrow \Delta_{\text{msg}}(u - u_{\text{sent}})$
beacon already scheduled?
yes
no
$t_{\text{sent}} + \Delta_t < t_{\text{sched}}$
no
yes
$t_{\text{sched}} \leftarrow$ NOW $+ \Delta_t$
$t_{\text{sched}} \leftarrow \max(t_{\text{sent}} + \Delta_t, \text{NOW})$
schedule(ONSEND, t_{sched})
(a)
ONSEND
try ← MAXRETRIES
last vehicle?
yes
no
timeout scheduled?
yes
cancelAckTimeout()
no
scheduleAckTimeout()
$(u, \dot{x}) \leftarrow$ getVehicleData()
sendBeacon(u, $\dot{x}$, ackMap)
$u_{\text{sent}} \leftarrow u$
$t_{\text{sent}} \leftarrow$ NOW
STOP
(b)
ONACKTIMEOUT
try == 0?
yes
notifyEmergency()
no
try ← try - 1
resendLastBeacon()
scheduleAckTimeout()
STOP
(c)

Figure 4.10 – Flow charts for the main protocol loop, sending, and ack timeout procedures. Copyright © 2015 IEEE.

choose. In particular, by using double precision floats (for u and $\dot{x}$) and 64-bit integers (for the sequence numbers), a 20-car platoon would result in an application layer payload of 200 B. Given that 32-bit sized integers and floating points are precise enough, with the same payload size we can support up to 45 vehicles. As a further improvement, we add

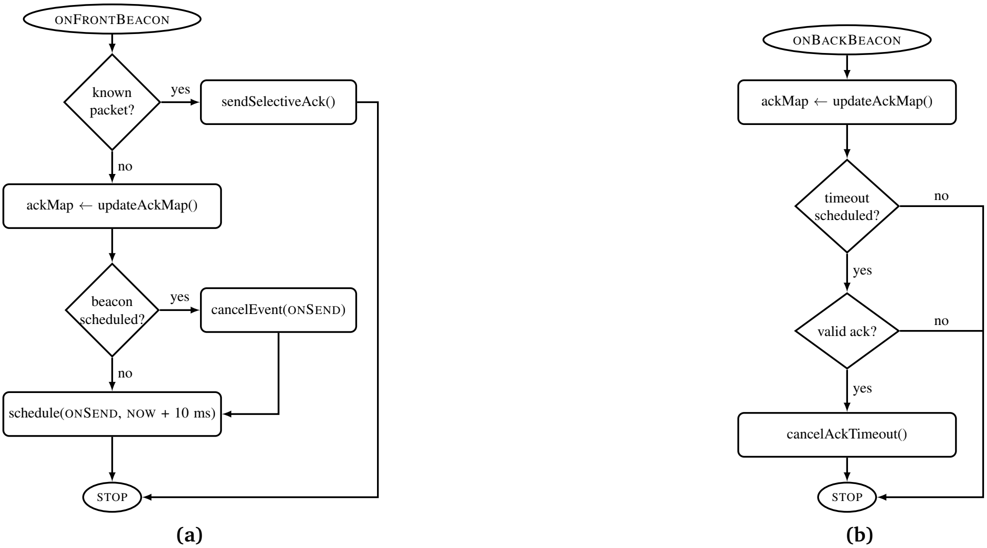


Figure 4.11 – Flow charts for message reception handling. Copyright © 2015 IEEE.

the static transmit power assignment proposed in Section 4.1 which provides enormous benefits.

We show protocol's flow diagrams in Figures 4.10 and 4.11. In the main loop (Figure 4.10a) the protocol invokes the Δ_{msg} function (Equation (4.5)) every 10 ms (the CACC sampling time). With the value obtained from the function, it decides when (and if) to schedule a new beacon. When sending a new beacon (Figure 4.10b), the protocol simply needs to reset the number of attempts and to schedule the timeout for the acknowledgement. This step is unnecessary for the last vehicle, as there is no other platooning member behind it. For cooperative awareness reasons, however, it still broadcasts its data packet.

Figure 4.10c handles the acknowledgement timeout event. In such a case, the protocol decrements the number of remaining attempts and re-sends the beacon. When the number of attempts reaches zero, the protocol notifies the network problem to the application layer. The application then needs to decide the best action to be taken to ensure the safety of the passengers. This decision is non trivial and depends on the conditions, the type of fault, the available backup communication technologies, etc. We find studies dealing with fault detection and recovery, including the handling of mechanical or network faults [39,85,87]. For example, in [85] the authors substitute the desired acceleration u_{i-1} obtained through wireless communication with a radar-based estimate when the former becomes unavailable. Another possibility is to use a backup communication technology like Visible Light Communication (VLC), as we propose in [102]. We believe this is a large problem on its own and we thus explicitly disregard it in this thesis. When such a condition is triggered in our simulator, we log the failure and stop the simulation.

When the protocol receives a beacon, if this is coming from the vehicle directly in front (Figure 4.11a), then we first check if it is a duplicate packet. This happens when a beacon is correctly received, but the corresponding acknowledgement is lost: In such a case, the protocol sends a selective acknowledgement. If the protocol, instead, recognizes a new packet, it updates the acknowledgements map and schedules an ONSEND event in 10 ms. According to the piggyback mechanism, this new scheduled beacon will also acknowledge the latest received one.

The final possible event (Figure 4.11b) is the reception of a beacon from any of the following vehicles. In this case, the protocol updates the acknowledgements map and, if the packet contains a valid acknowledgement, it cancels the timeout.

4.4.2 Performance Analysis

The scenario in which we test our proposal is a crowded traffic jam. To be coherent with the analysis in Sections 4.3.1 to 4.3.3 we again consider a 4-lane freeway, 20-car platoons, and a total number of cars of 160, 320 and 640. This time, however, we also need to consider vehicle dynamics to test the effectiveness of our approach. To this purpose, we add a “jamming car” at the head of each lane. The role of this vehicle is to generate a traffic shockwave by continuously changing its cruising speed every 30 s. To increase the dynamics also from a networking perspective, the vehicles generating the shockwaves are not perfectly synchronized. This way, platoons on different lanes always travel close one another, but misaligning over time. Leaders are driven by a standard ACC while all the followers by the PATH’s CACC.

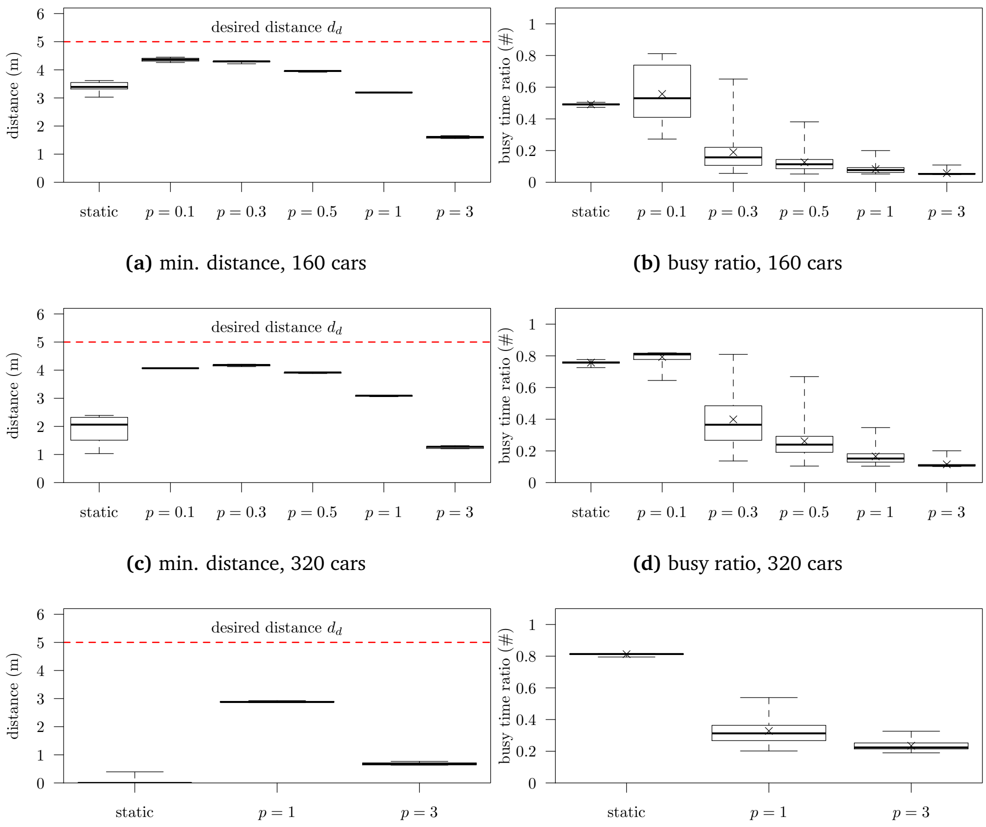
We consider a “harsh” and a “gentle” traffic scenarios. The harsh scenario is more dangerous and demanding: In the harsh setup the speed of the jammers changes between 130 km/h and 30 km/h with a deceleration of 7 m/s^2 and an acceleration of 1.5 m/s^2. Conversely, the idea of the gentle scenario is to simulate a more common slow down caused by, for instance, slower overtaking vehicles. In this latter case, jamming vehicles switch their speed from 130 km/h to 110 km/h and back with a deceleration of 3 m/s^2 and an acceleration of 1.5 m/s^2. Overall the simulation time is 180 s, thus repeating three times the traffic jam cycle. For statistical confidence, each configuration is repeated 10 times.

Table 4.3 – Network and road traffic simulation parameters. Copyright © 2015 IEEE.

	Parameter	Value
communication	Path loss model	Free space ($\alpha = 2.0$)
	Fading model	Nakagami ($m = 3$)
	PHY model	IEEE 802.11p
	MAC model	1609.4 single channel (CCH)
	Frequency	5.89 GHz
	Bitrate	6 Mbit/s (QPSK R = $^1/_2$)
	CCA-threshold	−65 dBm
	Access category	AC_VI
	MSDU size	200 B
	Transmit power	20 dBm and 0 dBm
	max_{bi}, min_{bi}, $\Delta_{u_{\text{max}}}$	1 s, 0.01 s, 2 m/s^2
	p	0.1, 0.3, 0.5, 1 and 3
mobility	Max speed	130 km/h
	Min speed (harsh/gentle)	30 km/h and 110 km/h
	Deceleration (harsh/gentle)	7 m/s^2 and 3 m/s^2
	Acceleration	1.5 m/s^2
	Platoon size	20 cars
	Number of cars	160, 320 and 640
(C)ACC	Engine lag τ	0.5 s
	CACC's C_1, ω_n, ξ, d_d	0.5, 0.2 Hz, 1, 5 m
	ACC's T, λ	1.2 s, 0.1

We test *Jerk Beaconing* against STB and STBP, which both use a beacon frequency of 10 Hz. The configuration parameters for *Jerk Beaconing* are $\text{max}_{\text{bi}} = 1\,\text{s}$, $\text{min}_{\text{bi}} = 0.01\,\text{s}$, $\Delta_{u_{\text{max}}} = 2\,\text{m/s}^2$. Moreover, we choose different values for the reactivity parameter p, i.e., 0.1, 0.3, 0.5, 1, and 3. We set the maximum number of retries to 5, the acknowledgement timeout to 50 ms, and we enable the prediction of leader and front vehicle speed as in Equation (4.3). Finally, to allow a fair comparison between *Jerk Beaconing* and the static approaches, we test *Jerk Beaconing* with and without transmit power control, using the same power assignments we give to STB and STBP. Table 4.3 summarizes all simulation parameters.

We make use of three different metrics to evaluate and compare the protocols. To have an idea of how safe the protocols are, we compute the minimum distance between any pair of consecutive vehicles in the simulation, for all repetitions. Then, we consider channel busy ratio as in Section 4.3.1 to understand the amount of network load generated by each protocol setup. Data is sampled at 1 Hz by each vehicle and plot in boxplots as in Section 4.3.1. The final metric is the beacon inter-arrival time, for which we compute the ECDF. The idea is to understand how much resources each protocol saves compared to the others. The beacon inter-arrival time metric, however, needs to be analyzed together with minimum distance and channel busy ratio, because high values for such a metric might not necessarily indicate resource saving, but severe packet losses as well.



(a) min. distance, 160 cars

(b) busy ratio, 160 cars

(c) min. distance, 320 cars

(d) busy ratio, 320 cars

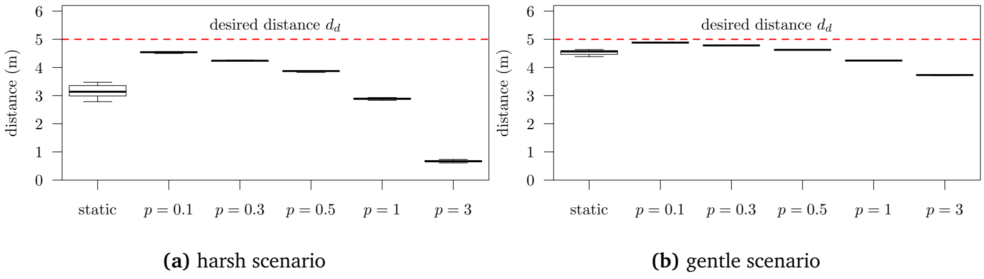
(e) min. distance, 640 cars

(f) busy ratio, 640 cars

Figure 4.12 – Minimum distances and channel busy ratio for the different protocols when transmit power control is not employed, harsh scenario. Copyright © 2015 IEEE.

We begin the analysis by considering the scenario where transmit power control is disabled. Figure 4.12 shows minimum distance and busy ratio plots for the harsh scenario. The results confirm the findings in Sections 4.3.1 and 4.3.2, i.e., static beaconing without transmit power control is unable to support the CACC. In a low density scenario (160 cars), the network is not yet overloaded. By looking at minimum distances, indeed, there are no major safety concerns besides *Jerk Beaconing* for $p = 3$, where the minimum distance goes as low as 1.5 m.

Increasing the vehicle density (320 cars) causes the network to start suffering. The minimum distances for STB span between 1 m and 2.5 m, while for *Jerk Beaconing* ($p = 3$)



(a) harsh scenario **(b)** gentle scenario

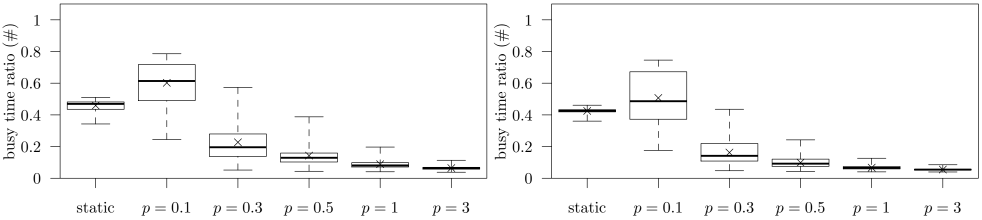
Figure 4.13 – Minimum distances for the different protocols, 640 cars. Copyright © 2015 IEEE.

they concentrate around 1 m. *Jerk Beaconing* for other values of p is still safe enough, but for $p = 0.1$ the majority of the simulations were stopped due to the acknowledgement timeout, meaning that the network was overloaded. This issue is highlighted by the channel busy ratio plot in Figure 4.12d. STB and *Jerk Beaconing* for $p = 0.1$ definitely overload the network, explaining why STB results in an unacceptable safety performance and why *Jerk Beaconing* declares network failure.

In the highest density scenario (640 cars), the network load is unsustainable. STB results in crashes in more than 75 % of the cases, while *Jerk Beaconing* for values of p lower than 1 always declares network failure, which is the reason why we plot no results in such cases. By keeping the network load under control, *Jerk Beaconing* for $p = 3$ never results in crashes, but it still lacks safety performance. For $p = 1$, instead, *Jerk Beaconing* still properly works, even though minimum distance is around 3 m. Figure 4.12f shows how, opposed to STB, *Jerk Beaconing* does not overload the network even in the highest density scenario.

By enabling transmit power control, we obtain very similar results independently from the scenario density. The interference range is indeed reduced to the minimum by the transmit power policy, and the high, standard compliant CCA-thresholds (—65 dBm) makes vehicle ignore the interference generated by farther ones, maximizing spatial reuse. For this reason we only show the results of the highest density scenario (640 cars). Furthermore, thanks to the reliable protocol, *Jerk Beaconing* declared network failure only in very rare cases.

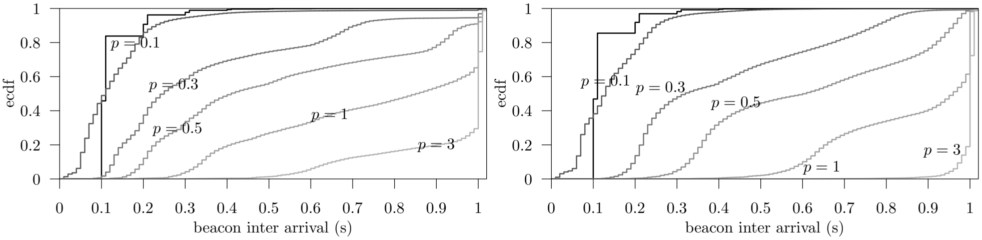
Figure 4.13a shows the results for the harsh scenario in terms of minimum distance. The plot shows that all protocols, including STBP, benefit from the usage of transmit power control. Indeed, no protocol cause an accident. If we compare the approaches, we see that *Jerk Beaconing* for p equal to 0.1, 0.3 and 0.5 is safer than STBP, which in



(a) harsh scenario

(b) gentle scenario

Figure 4.14 – Channel busy time for the different protocols, 640 cars. Copyright © 2015 IEEE.



(a) harsh scenario

(b) gentle scenario

Figure 4.15 – Empirical CDF of beacon inter-arrival times for the different protocols, 640 cars. Copyright © 2015 IEEE.

turn is safer than *Jerk Beaconing* when choosing $p = 1$ or $p = 3$. As in the analysis without transmit power control, when we set $p = 3$, *Jerk Beaconing* is too conservative and tries to maximize resource saving at the expenses of safety. In the gentle scenario (Figure 4.13b) we have qualitatively similar results, but quantitatively better because of the lower reactivity demand.

Figure 4.14 compares the protocol from the network perspective, i.e., in terms of channel busy ratio. In particular, the figure shows the price to pay for the safety of *Jerk Beaconing*, $p = 0.1$. In terms of vehicle dynamics, *Jerk Beaconing* for $p = 0.1$ performs better than any other protocol, but it can overload the network and uses, in the considered scenarios, more resources than STBP. For values of p greater than 0.1, *Jerk Beaconing* on average saves at least half of the resources, which could potentially be used by other applications. By comparing Figures 4.14a and 4.14b, we see qualitatively similar results for the harsh and for the gentle scenarios: Quantitatively, in the gentle scenario all protocols use slightly less channel resources. This is the case for STBP as well because of the different cruising speeds, which cause the density of the vehicles to change. Indeed, in the harsh scenario, during the slow cruising phase (i.e., at 30 km/h), the ACC maintains an inter-platoon gap of only 10 m. In the gentle scenario, instead, this distance is never smaller than 36 m. A further confirmation of this fact is given by the variance in channel load for STBP, which, in the gentle scenario, is smaller.

We now consider the beacon inter-arrival distribution of leader messages (Figure 4.15). In the graph, the black, steep line plots the ECDF of STBP: In the majority of the cases, STBP delivers the packets within 100 ms. Some exceptions are due to packet losses. Furthermore, Figure 4.15 explains why, for $p = 0.1$, *Jerk Beaconing* loads the network more than STBP. In the 50 % of the cases and for the harsh scenario, the inter-arrival time is smaller than 100 ms. For the gentle scenario, the percentage reduces to 40 %, but still enough to cause the overload. As highlighted in Figure 4.14, higher values of the p parameter result in an efficient usage of network resources. For $p = 0.5$, in the harsh scenario *Jerk Beaconing* uses a beacon interval larger than 400 ms 50 % of the time. In the gentle scenario, i.e., where *Jerk Beaconing* detects there is no need to use more network resources, the inter-arrival is no smaller than 650 ms in the 50 % of the cases.

In summary, the plots in Figures 4.13 to 4.15 show that for $p = 0.5$, *Jerk Beaconing* is safer than STBP using at most 25 % of the resources, proving the potential of dynamic approaches for platooning control systems.

4.5 Conclusion

In this chapter we proposed and analyzed different beaconing strategies for a platooning system. We started by considering the application layer requirements, i.e., specifically taking into account that the CACC needs to obtain data from the platoon leader and the vehicle directly in front. Following this considerations we designed a beaconing protocol, which exploits coordination through a TDMA like approach to reduce intra-platoon channel contention and transmit power control to reduce the interference domain and increase spatial channel reuse. The results, compared to two dynamics approaches (ETSI DCC and DynB), proved the effectiveness of taking into account application layer requirements. In particular, we showed that transmit power control is essential and provides the highest benefit. The slotted approach, instead, provides only a slight improvement, but statistically irrelevant. Overall, the proposed solutions timely deliver (i.e., within 200 ms) control information 90 % of the time.

After observing that the behavior of the controller depends on the dynamics of the vehicles, we then relaxed the 10 Hz beaconing requirement assumption and developed *Jerk Beaconing*. *Jerk Beaconing* computes the beacon interval depending on the dynamics of the vehicle and uses a prediction mechanism to interpolate missing data points. Moreover, *Jerk Beaconing* exploits a TDMA like approach and piggybacking of the acknowledgements to ensure data delivery. The proposed approach showed not only enormous benefits in terms of network resource saving, but it has also shown to be safer than a static, 10 Hz approach. Even in the most demanding scenario, *Jerk Beaconing* (for a sensitivity value of 0.5) never resulted in a minimum distance lower than 4 m using only 25 % of the resources used by the static approach.

Chapter 5

A Consensus-based Controller

In this chapter we discuss the platooning controller we developed together with the colleagues at the University of Naples. The controller is based on the idea of “consensus”, and represents a change in perspective in the design of control systems for platooning. In particular, while the communication topology of common Cooperative Adaptive Cruise Controls (CACCs) is pre-established, i.e., front-vehicle only in [84], or leader-front in the PATH project [88], in our work it can be reconfigured at run time depending on requirements or network characteristics. This chapter is based on the following papers:

- S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno, “A Consensus-based Approach for Platooning with Inter-Vehicular Communications,” in *34th IEEE Conference on Computer Communications (INFOCOM 2015)*. Hong Kong: IEEE, April 2015, pp. 1158–1166. My contribution was providing the underlying fundamentals, enabling the implementation and the realistic analysis of the controller in PLEXE.

- S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno, “A Consensus-based Approach for Platooning with Inter-Vehicular Communications and its Validation in Realistic Scenarios,” in *IEEE Transactions on Vehicular Technology*. IEEE, 2015, submitted for peer review. My contribution was the design and the implementation of the realistic engine and braking model described in Section 2.3 as a new feature for the PLEXE simulator.

5.1 Background and Motivation

The aim of the work in [90, 91] is to take into consideration the communication-induced impairments in the design of the controller. The previous chapters described different platoon control designs, including parameter tuning and spacing policies to achieve and ensure string-stability. In the literature we find several works discussing stability issues of platooning systems. As an example, in [46] the authors show that predecessor-following architectures are very sensitive to external disturbances and to the number of vehicles in the platoon, and [89] shows that a spacing policy based on time headway helps in stabilizing the platoon. When proving string-stability properties of a system, the common assumption is that Inter-Vehicle Communication (IVC) introduce no delay, i.e., ideal communication [111]. In practice, delays and failures introduced by the communication medium highly affect the stability of the platoon [75].

Recently, simulation-based studies started to consider this problem in their analysis and to propose protocols that can mitigate the effects [36]. However, given that control algorithms do not explicitly take into account delays in the design phase, most works investigate the effects assuming a constant communication delay [35, 75]. The usage of Veins- and PLEXE-like simulation tools permits to consider more realistic communication impairments, as done in [73] for the CACC designed in [80]. Furthermore, the logical communication topology among the vehicles in a platoon is decided at design time. In a real-world environment, however, it might not be realizable due to communication problems, or we might want to switch to a different one to change the behavior and/or the characteristics of the controller. When we consider generic topologies for the control of a platoon, we fall into the category of multi-agent systems control [81]. We can find examples of platooning control in this recent research direction in [112, 121], or other works focusing on properties and performance of this kind of control systems when the topology is time-varying [74, 125].

The idea of the controller we propose is to consider platooning as a high order consensus problem, which accounts for a non-fixed communication topology with time varying delays. Distributed consensus is one way to approach the design of networked control systems. In a consensus problem, all nodes in a network should reach an agreement, i.e., at steady state they should all converge to the same goal [50]. In the particular case of vehicle platooning we want to have the same distance between pairs of consecutive vehicles [121]. The work extends the analysis in [27] by embedding a constant time-gap policy plus a standstill requirement [80], and by considering a communication delay, which is not unique among all vehicles. We test the proposed approach using PLEXE, thus considering realistic networking and engine/dynamics models.

5.2 Consensus-based Control

In the scope of multi-agent control systems we model the network as a graph, where every node is a vehicle. We thus represent a platoon made of N vehicles as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order $|\mathcal{V}| = N$, with $\mathcal{V} = \{1, \ldots, N\}$ being the set of nodes. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges, while $\mathcal{A} = \left[a_{ij}\right]_{N \times N}$ is the corresponding adjacency matrix, i.e.,

$$a_{ij} = \begin{cases} 1 & \text{if } (j,i) \in \mathcal{E} \\ 0 & \text{if } (j,i) \notin \mathcal{E}. \end{cases} \tag{5.1}$$

Notice that $a_{ij} = 1$ indicates a communication link from vehicle j to vehicle i, and not vice versa. $\mathcal{A}$ (and thus $\mathcal{E}$) represents the logical topology, i.e., whether a vehicle should consider or not the information received from another vehicle when computing the control action, and does not necessarily represent the actual network status. For example, in a small 4-car platoon, we might have an all-to-all wireless connectivity but we might want to consider front vehicle information only. Following this reasoning, $a_{ij} = 1 \not\Longrightarrow a_{ji} = 1$. Moreover, $a_{ii} = 0$, i.e., there are no self loops. The leading vehicle is not included in $\mathcal{G}$ and it is treated as an additional agent of index zero.

$\mathcal{A}$ makes the control topology configurable. For example, to use the control topology of the controller developed by Ploeg et al. [84], the a_{ij} matrix for a 4-car platoon would be

$$a_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{5.2}$$

while for the PATH CACC, a_{ij} would be

$$a_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \tag{5.3}$$

We define the dynamics of the i-th vehicle ($i = 1, \ldots, N$) as:

$$\dot{x}_i(t) = v_i(t) \tag{5.4}$$

$$\dot{v}_i(t) = \ddot{x}_i(t) = \frac{1}{M_i} u_i(t), \tag{5.5}$$

with $x_i(t)$ and $v_i(t)$ being the absolute position and speed of the i-th vehicle respectively. In the first definition of the problem we assume that

$$i < j \iff x_i(t) > x_j(t), \quad \dot{x}_i(t) \geq 0, \tag{5.6}$$

meaning that vehicles drive on a line with a positive speed and a monotonically increasing position. We then relax this assumption and provide the control formula for a “real world” implementation. M_i represents the mass and u_i the control input (in this particular case, the propelling force) to be chosen to achieve the control goal. We instead define leader vehicle dynamics as:

$$\dot{x}_0(t) = v_0 \tag{5.7}$$

$$\dot{v}_0(t) = \ddot{x}_0(t) = 0. \tag{5.8}$$

Given Equations (5.4) and (5.7), to maintain a desired inter-vehicle distance and the same speed we define the following high-order consensus problem:

$$x_i(t) \to \frac{1}{\Delta_i} \left\{ \sum_{j=0}^{N} a_{ij} \cdot \left(x_j(t) + d_{ij} \right) \right\} \tag{5.9}$$

$$\dot{x}_i(t) \to v_0. \tag{5.10}$$

In Equation (5.9), d_{ij} is the desired distance between i and j and $\Delta_i = \sum_{j=0}^{N} a_{ij}$ is the in-degree of node i. In this case, a_{ij} is defined for $i = 1, \ldots, N$ and for $j = 0, \ldots, N$, as vehicles receive data from the leader as well, but the leader is not influenced by the followers. The desired distance term is defined as

$$d_{ij} = h_{ij} v_0 + d_{ij}^{\text{st}}, \tag{5.11}$$

where h_{ij} is the time headway between vehicles i and j, and d_{ij}^{st} is the standstill distance between the front bumpers, which includes the length of the vehicles as well. To formally define h_{ij} and d_{ij}^{st}, we start by defining the base cases, i.e., we choose and fix the values for consecutive vehicles $h_{i,i+1}$ and $d^{\mathrm{st}}_{i,i+1}$:

$$h_{i,i+1} > 0, \quad i = 0, \dots, N-1$$
$$d^{\mathrm{st}}_{i,i+1} = l_i + d^*_{i,i+1} > 0, \quad i = 0, \dots, N-1$$

where l_i is the length of the i-th vehicle and $d^*_{i,i+1}$ is the bumper-to-bumper standstill distance. We now define the inverse cases as

$$h_{i+1,i} = -h_{i,i+1}, \quad d^{\mathrm{st}}_{i+1,i} = -d^{\mathrm{st}}_{i,i+1}, \quad i = 0, \dots, N-1.$$

Finally, we compute the general cases as

$$h_{i,j} = \begin{cases} \sum_{k=j}^{i-1} h_{k+1,k} & \text{if } i > j \\ -h_{j,i} & \text{if } i < j \end{cases} \tag{5.12}$$

and

$$d^{\mathrm{st}}_{i,j} = \begin{cases} \sum_{k=j}^{i-1} d^{\mathrm{st}}_{k+1,k} & \text{if } i > j \\ -d^{\mathrm{st}}_{j,i} & \text{if } i < j. \end{cases} \tag{5.13}$$

We solve the consensus problem in Equation (5.9) using a decentralized control action, which accounts for the spacing policy and the time-varying communication delays as:

$$\begin{aligned} u_i(t) = -b_i \underbrace{\left(\dot{x}_i(t) - \dot{x}_0(t - \tau_{i0}(t))\right)}_{\text{speed error}} - \\ \frac{1}{\Delta_i} \sum_{j=0}^{N} k_{ij} a_{ij} \underbrace{\left(x_i(t) - x_j(t - \tau_{ij}(t)) - \tau_{ij}(t) \cdot \dot{x}_0(t - \tau_{i0}(t)) - d_{ij}\right)}_{\text{distance error}}. \end{aligned} \tag{5.14}$$

The first part of the formula $(-b_i(\dot{x}_i(t) - \dot{x}_0(t - \tau_{i0}(t))))$ computes the speed error with respect to the leading vehicle. The controller assumes all positions and speeds to be obtained via wireless communication. For this reason, the speed of the leader is not the speed at time t, but at the time the information was sent. $\tau_{ij}(t)$ is indeed the elapsed time from the moment vehicle i received a data packet from vehicle j up until time t. In the second part, the controller computes the position error. $x_i(t) - x_j(t - \tau_{ij}(t)) - \tau_{ij}(t) \cdot \dot{x}_0(t - \tau_{i0}(t))$ is the GPS distance between vehicles j and i. $\tau_{ij}(t) \cdot \dot{x}_0(t - \tau_{i0}(t))$ is a compensation term for the outdated position of vehicle j $(x_j(t - \tau_{ij}(t)))$ which assumes all vehicles should travel at leader's speed. Finally, d_{ij} is the desired distance (Equation (5.11)).

The remaining parameters are:

- $b \in \mathbb{R}^{N+1}$: Vector of gains for speed error;
- $a \in \{0,1\}^{(N+1)\times(N+1)}$: Matrix that models the control topology;
- $k \in \mathbb{R}^{(N+1)\times(N+1)}$: Matrix of position error gains;
- Δ_i: In-degree of vehicle i ($\sum_{j=0}^{N} a_{ij}$);
- N: Number of followers in the platoon.

Equation (5.14) is defined assuming that vehicles travel in a mono-dimensional space with a monotonically increasing position (Equation (5.6)). For a real world implementation we can relax this assumption by considering a two- or three-dimensional coordinate system, i.e., $x_i(t)$ becomes $\boldsymbol{x}_i(t)$ and $\dot{x}_i(t)$ becomes $\dot{\boldsymbol{x}}_i(t)$. Equation (5.14) can thus be recast as

$$u_i(t) = -b_i \underbrace{(\|\dot{\boldsymbol{x}}_i(t)\|_2 - \|\dot{\boldsymbol{x}}_0(t-\tau_{i0}(t))\|_2)}_{\text{speed error}} -$$

$$\frac{1}{\Delta_i}\sum_{j=0}^{N} k_{ij}a_{ij}\underbrace{\left(\|\boldsymbol{x}_i(t)-\boldsymbol{x}_j(t)-\tau_{ij}(t)\cdot\dot{\boldsymbol{x}}_0(t-\tau_{i0}(t))\|_2\cdot\operatorname{sgn}(j-i)-d_{ij}\right)}_{\text{distance error}}. \quad (5.15)$$

5.3 Evaluation

5.3.1 Network and Vehicular Traffic Scenario

To test Equation (5.14), we implemented the controller in PLEXE together with different simulation scenarios. For what concerns channel modeling, we first test the controller in ideal conditions, i.e., with no packet loss. We then consider a realistic network setup in a freeway scenario where human-driven vehicles generate interference by transmitting periodic Cooperative Awareness Messages (CAMs). In this realistic setup, we consider a free-space path loss model together with Nakagami-m distributed fading. We configure PHY and MAC layer models as in Chapter 4, i.e., using IEEE 802.11p/1609.4 standards, and consider a 10 Hz beaconing frequency.

For the realistic scenario, we design a bi-directional 4-lane freeway where different types of human-driven vehicles travel (Figure 5.1). Standard vehicles travel on the three rightmost lanes in both directions, while the fourth is reserved to platooning vehicles. We consider normal cars, vans, and trucks in different percentages and running at different

Table 5.1 – Network simulation parameters.

Parameter	Value
Path loss model	Free space ($\alpha = 2.0$)
Fading model	Nakagami-m ($m = 3$)
PHY/MAC model	IEEE 802.11p/1609.4 single channel (CCH)
Frequency	5.89 GHz
Bitrate	6 Mbit/s (QPSK R = 1/2)
Access category	AC_VI
MSDU size	200 B (byte)
Transmit power	20 dBm
Beacon frequency	10 Hz

Table 5.2 – Traffic simulation parameters for the realistic scenario. Copyright © 2015 IEEE.

Freeway length	10 km
Lanes	4 (two-way)
Cars percentage (length 4 m)	50 %
Trucks percentage (length 20 m)	20 %
Vans percentage (length 5 m)	30 %
Inter-vehicle time	$\sim \exp(0.7276/\mathrm{s})$ ($\mathbb{E}[X] = 1.374\,\mathrm{s}$ [123]
Cars' speed	$\sim U(100\,\mathrm{km/h}, 160\,\mathrm{km/h})$
Trucks' speed	80 km/h
Vans' speed	100 km/h
Platoon size	8 and 16 cars
Platooning car max acceleration	$2.3\,\mathrm{m/s^2}$
Platooning car mass	1460 kg
Platooning car length l_i	4 m
Headway time h_{ij}	0.8 s
Control gains k_{ij}	$k_{10} = 460$, $k_{i0} = 80\,(i \neq 0, i \neq 1)$ $k_{i,i-1} = 860$, $k_{ij} = 0$ otherwise
Control gains b_i	$b_i = 1800$
Distance at standstill d^{st}	15 m
Freeway fill-up time	500 s
Network warm-up time	10 s
Data recording time	50 s

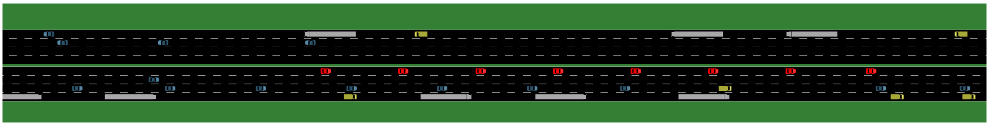


Figure 5.1 – Screenshot of the realistic scenario. Human-driven vehicles in white, blue, and yellow, and platooning cars in red on the left-most lane. Copyright © 2015 IEEE.

speeds. We inject human-driven vehicles waiting for an exponentially distributed inter-vehicle time [123]. The simulation waits 500 s to fill up the freeway with human-driven vehicles, and then injects the platoon in the middle of the freeway. Data recording starts after a warm-up time. Tables 5.1 and 5.2 list all network and mobility simulation parameters.

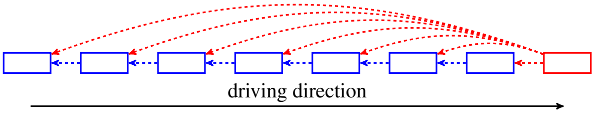
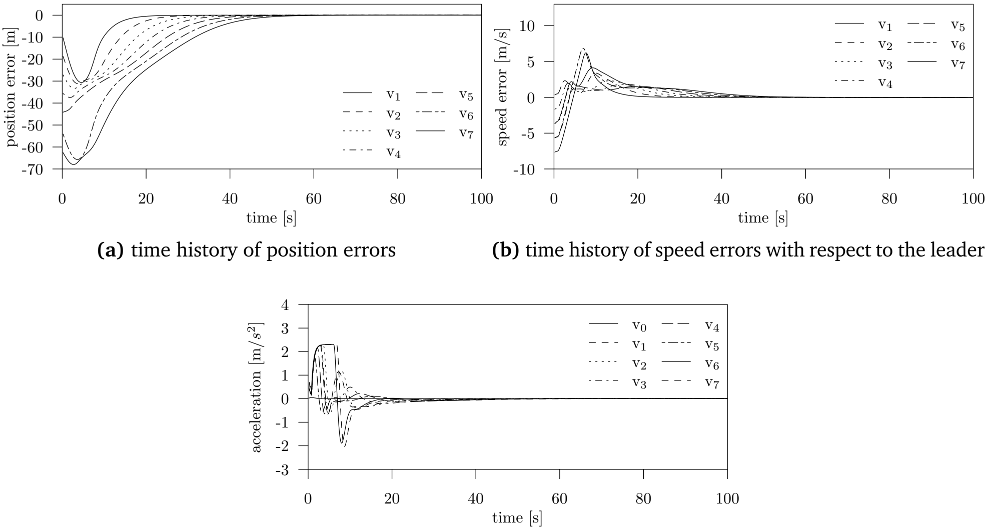


Figure 5.2 – Control topology selected for the simulations. Copyright © 2015 IEEE.



(a) time history of position errors

(b) time history of speed errors with respect to the leader

(c) time history of accelerations

Figure 5.3 – Basic convergence analysis with $v_0 = 100\,\mathrm{km/h}$, $N + 1 = 8$ vehicles. Platoon creation and maintenance. Copyright © 2015 IEEE.

We test the performance in terms of stability and robustness by analyzing the behavior of the controller in different maneuvers. In a first maneuver, we simply test whether the controller reaches the consensus goal: In this case, the leader simply drives at constant speed and the followers try to reach the desired inter-vehicle spacing. We then consider a leader tracking maneuver, i.e., the leader accelerates or decelerates to change its cruising speed and the followers should correctly track its behavior. Finally, we disturb the platoon by changing leader's speed with a sinusoidal trend.

We consider a leader/front topology as considered in [78] to compare it with the PATH's CACC. Figure 5.2 depicts the control topology. Algorithm convergence, however, is not bound to the chosen topology, and would be reached also when considering classical predecessor-following architectures like in [84].

5.3.2 Basic Convergence Analysis

To test basic convergence (i.e., reaching of consensus) we run a platoon of 7 cars plus a leader and consider no packet losses. Figure 5.3 shows the results for this scenario, confirming the ability of the consensus controller of creating and maintaining the platoon. Despite the presence of network delays during the communication, all vehicles reach the consensus converging to the desired spacing and leader speed. Even if the consensus problem is formulated for constant leader speed v_0, the algorithm correctly tracks the disturbances caused by leader changes in dynamics. Figure 5.4 shows a first example: The leader accelerates from standstill up to 90 km/h with a constant acceleration of 0.5 m/s^2 and all the followers correctly track its behavior reaching the consensus speed. Similarly, in Figure 5.5 decelerates from 100 km/h to a complete stop simulating a braking maneuver. All vehicles safely come to a stop and converge to the stand-still distance of 15 m. Finally, we introduce a sinusoidal disturbance to the leader's motion to show the string stability properties of the controller. In particular, we disturb leader's speed by adding the following sinusoid:

$$\delta(t) = A\cos\left(\frac{6}{100}\pi t\right), \quad A = 2.7\,\mathrm{m/s}. \tag{5.16}$$

Figure 5.6 shows how bumper to bumper distances are effectively attenuated downstream the string of vehicles.

We further analyze the controller in similar conditions, but considering a platoon of 16 vehicles using a constant spacing policy, i.e., traveling with a fixed inter-vehicle distance of 5 m. Changing the conditions does not change the qualitative results: The controller reaches the consensus for both distance (Figure 5.7a) and speed (Figure 5.7b). Moreover, the controller is string stable even for a constant gap policy (Figure 5.7c).

We now relax the idealistic network assumptions and consider the realistic freeway scenario depicted in Figure 5.1, where human-driven vehicles disturb the communication by beaconing at 10 Hz. We first consider a leader tracking maneuver, where the leader accelerates from 80 km/h to 130 km/h with an acceleration of 1.5 m/s^2. Figure 5.8 shows the speed of the vehicles over time during the maneuver. Despite the interferences, the maneuver is correctly executed with minor differences to Figure 5.4. We then consider a sinusoidal disturbance: The leader cruises at an average speed of 110 km/h, oscillates at a frequency of 0.2 Hz. In Figure 5.9 we plot bumper to bumper distances over time, showing how the controller successfully attenuates the distance error towards the end of the platoon. In particular, for the third vehicle the disturbance becomes already negligible. Packet loss causes only minor glitches, which can hardly be spotted in the plot. By analyzing the data in detail, indeed, we discovered that vehicle 7 looses its reference position between 602 s

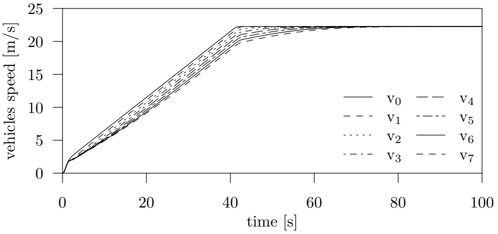


Figure 5.4 – Leader tracking maneuver: Time history of vehicle speeds. Copyright © 2015 IEEE.

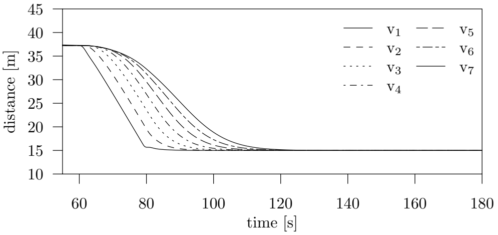


Figure 5.5 – Braking maneuver: Time history of bumper to bumper distances. Copyright © 2015 IEEE.

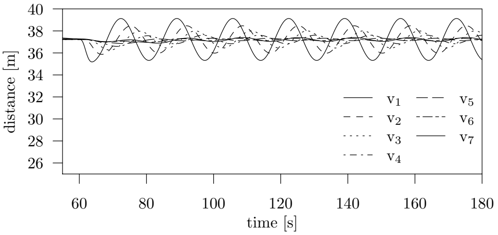
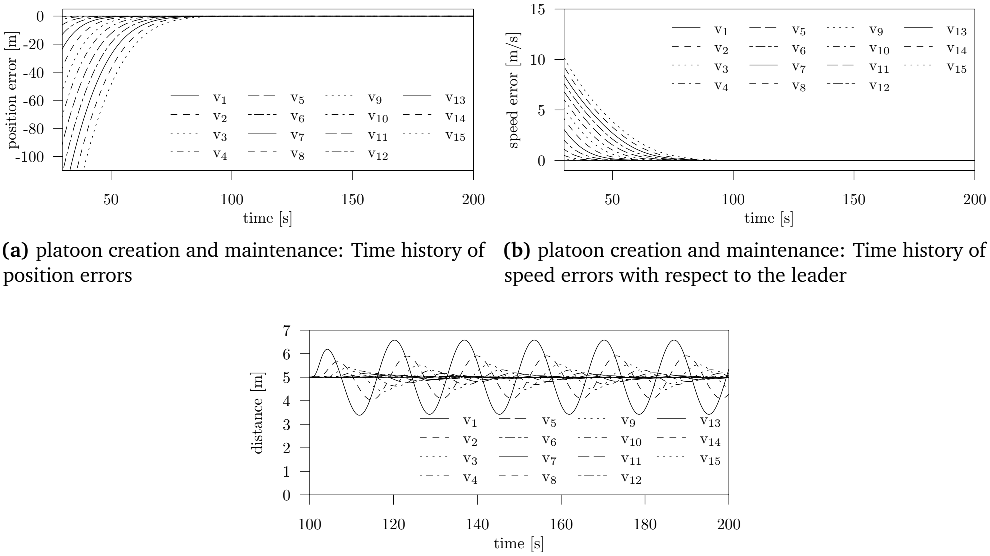


Figure 5.6 – Robustness with respect to a sinusoidal disturbance: Time history of bumper to bumper distances. Copyright © 2015 IEEE.

and 606 s. The error, however, is in the order of 20 cm, so the system can thus still be considered safe and robust even in the case of network congestion.

5.4 Analysis with Realistic Vehicle Dynamics

The performance of a platoon does not only depend on the control algorithm, but also on the actuation capability of the vehicles, which should correctly implement the control action computed by the CACC. In the hierarchical architecture defined in the PATH project [37,86] we have an upper-level controller (i.e., our control strategy) which sends the control input



(a) platoon creation and maintenance: Time history of position errors

(b) platoon creation and maintenance: Time history of speed errors with respect to the leader

(c) robustness with respect to a sinusoidal disturbance: Time history of bumper to bumper distances

Figure 5.7 – Dynamics for a platoon of $N+1=16$ vehicles. Copyright © 2015 IEEE.

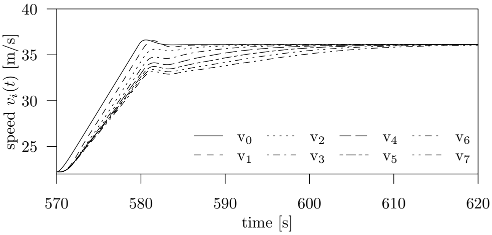


Figure 5.8 – Leader tracking maneuver in the realistic network scenario: Time history of vehicle speeds. Copyright © 2015 IEEE.

u_i to the lower-level controller. The lower-level controller then sends commands to the throttle or to the brake actuators to correctly track the desired acceleration. Still, a vehicle might not have enough “resources” to implement such command. In Section 2.3, indeed, we have seen that engine traction force and braking torque are limited, and that their limits depends on the mechanic and the aerodynamic properties of the vehicle. These practical limits might induce large transients and degrade the performance, so even if stability is theoretically proven we must test the robustness of the control algorithm in case of realistic, non-linear vehicle dynamics models.

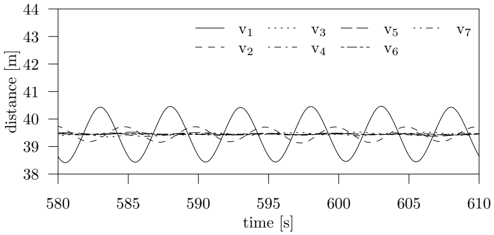
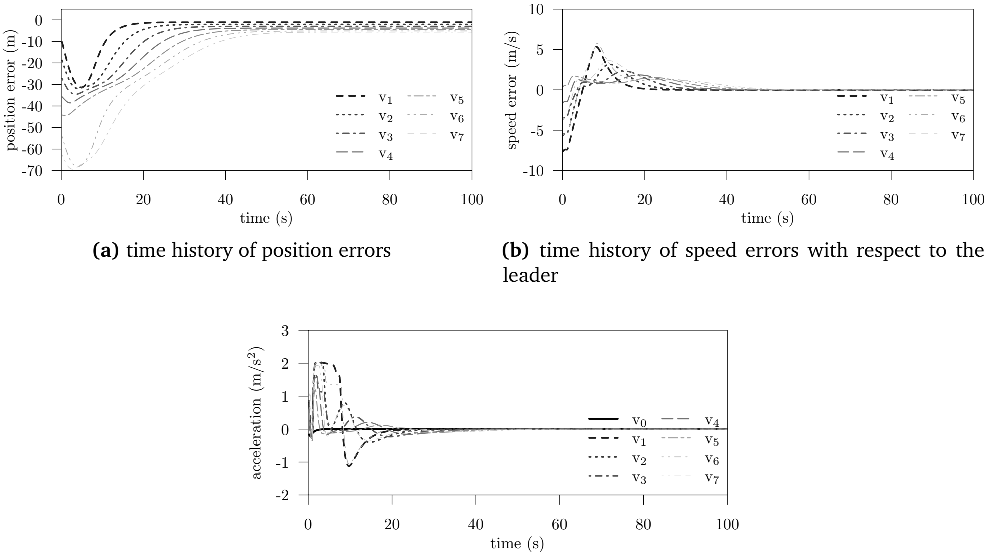


Figure 5.9 – Sinusoidal disturbance on leader motion in the realistic network scenario: Time history of bumper to bumper distances. Copyright © 2015 IEEE.



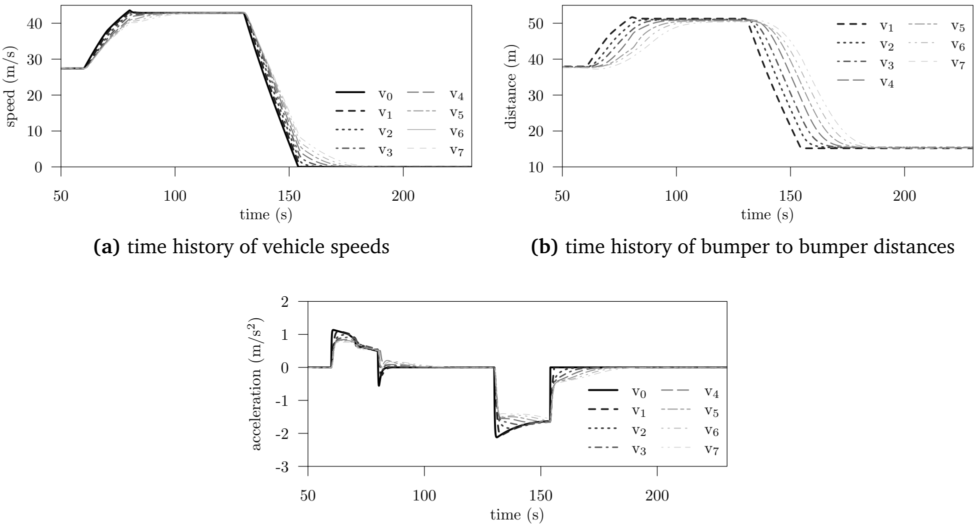
(a) time history of position errors

(b) time history of speed errors with respect to the leader

(c) time history of accelerations

Figure 5.10 – Platoon creation and maintaining: Convergence analysis with realistic vehicles models ($N+1=8$ and $v_0 = 100\,\mathrm{km/h}$).

The aim of this section is to test the performance of our consensus controller using the realistic vehicle model of Section 2.3. We simulate the leader tracking maneuver and the sinusoidal disturbance for a platoon of eight inhomogeneous cars. In particular, we consider the real characteristics of four commercial cars, i.e., an Audi A3 2.0 TDI, a BMW 1 Series 2.0, a Mini Cooper 1.6, and an Alfa Romeo 147 1.6 Twin Spark, and we randomly assign these characteristics to the vehicles of the platoon. We do not list here all the vehicle parameters for the sake of brevity, but they can easily be found online. We simulate the



(a) time history of vehicle speeds

(b) time history of bumper to bumper distances

(c) time history of accelerations

Figure 5.11 – Leader tracking maneuver with realistic vehicle model.

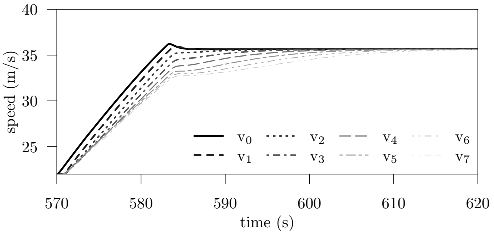


Figure 5.12 – Leader tracking maneuver with realistic network scenario and vehicle model. Time history of vehicle speeds.

same maneuvers we considered in the previous section, i.e., basic convergence, leader tracking, and sinusoidal disturbance.

We first analyze consensus in an ideal network setting, i.e., without interfering communication. Figure 5.10 shows the simulation results in this particular setting. Even when adding non-linear vehicle dynamics and when considering inhomogeneous platoons, the proposed approach is capable of forming and maintaining the platoon. In this case, we have a position error at steady state, but the error is negligible ($\simeq 1\,\%$) if we consider that the maximum traction effort is limited and that the vehicle is disturbed by air drag and rolling resistance.

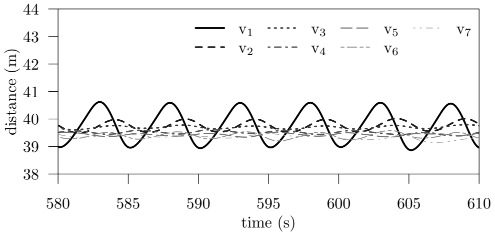


Figure 5.13 – Sinusoidal disturbance acting on leader motion with realistic network scenario and vehicle model. Time history of bumper to bumper distances.

In another experiment we analyze how the vehicles track the leader when the latter accelerates and then decelerates to a complete stop. Figure 5.11 shows speeds, distances, and accelerations of the vehicles over time in this particular maneuver. The controller achieves good performance even in this scenario despite the external disturbances and the limits of the vehicles. All vehicles successfully track leader speed and converge to standstill distance when stopping.

We conclude the analysis by considering leader tracking and sinusoidal disturbance maneuvers in the high traffic density scenario, where human-driven vehicles generate channel interference. In this setup, we repeat each experiment five times randomly changing the ordering of the vehicles to simulate different inhomogeneous platoons. In here, we show one of those five repetitions for the sake of brevity. In the leader tracking maneuver (Figure 5.12), all followers track leader's speed without overshooting the speed of their front vehicle. This means that the controller is robust and can counteract the impairments caused by real car dynamics, platoon inhomogeneities, and network congestion. Concerning the sinusoidal disturbance case, instead, Figure 5.13 shows that the platoon is stable even if the disturbance attenuation is not as effective as in the case of idealized vehicles (Figure 5.9). This final set of results confirm stability, convergence, and noise rejection properties.

5.5 Conclusion

In this chapter we develop a platooning controller based on the concept of consensus. The main feature of this controller is that its control topology is not fixed a-priori, but it can be configured at "run-time" depending on the desired characteristics or the actual network conditions. The control formula employs the speed of the leading vehicle as a reference and the positions of (potentially) all vehicles in the platoon to compute the required acceleration. We thoroughly tested the controller for different maneuvers (reaching the consensus, leader tracking, and sinusoidal disturbance), in different network conditions (ideal and heavily congested), and for different vehicle dynamics models. In all scenarios, the controller showed its robustness to impairments and external disturbances, as well as its convergence and string-stability characteristics, making it a valid candidate for real world deployment.

Chapter 6

Conclusion and Outlook

Vehicle platooning is expected to have an enormous impact on future Intelligent Transportation System (ITS) infrastructure by reducing traffic congestion, pollution, accidents, and the amount of wasted driving time. As platooning embraces different engineering/scientific disciplines, it is a very challenging application where some problems are still open. The research community clearly needs to address such issues before the deployment of platoon on real roads as design mistakes might lead to the loss of human lives.

In this thesis we focused on network communication problems for longitudinal platooning control in heavily dense scenario. As a first contribution we developed and released a platooning simulation tool that is freely available to download on the official website[5]. The tool is extremely versatile and, thanks to its detailed networking and vehicle dynamics model, it permits to investigate control systems, networking protocols, maneuvers and applications, as well as the impact of human-driven vehicles on the system. The simulator is easy to extend, permitting the user to easily implement new control algorithms or network applications.

Besides releasing this tool to the community, we used it to analyze the performance of an IEEE 802.11p-based communication system to support a platooning application. We proposed a set of static (i.e., periodic) communication protocols (STB, SLB, STBP, and SLBP) and analyzed them in different scenarios. First we have shown that using transmit power control is fundamental. Due to its random access nature, IEEE 802.11p highly suffers from network congestion, and transmit power control limits the interference domain and enables spatial reuse of the channel. The spatial reuse is further improved by using a high Clear Channel Assessment (CCA)-threshold as mandated by the IEEE 802.11 standard: A high

[5] `http://plexe.car2x.org`

value forces the MAC layer to ignore low-power interference due to farther neighbors when performing carrier sensing. We also investigated the impact of a Time Division Multiple Access (TDMA)-like communication to reduce intra-platoon channel contention. This mechanism, however, provides a statistically irrelevant advantage compared to the benefits of transmit power control. Overall, in 90 % of the cases, the SLBP protocol was capable of delivering the control information within 200 ms in the most demanding scenario.

For a more comprehensive analysis, we simulated an emergency braking scenario with different braking intensities and different beaconing frequencies, with the aim of understanding the requirements of the longitudinal control system. The results have shown that the requirements actually depend on the dynamics of the maneuver, i.e., the stronger the deceleration, the tighter the requirement. This led us to propose *Jerk Beaconing*, an approach that adapts the beacon interval to the dynamics of the vehicle and that uses a prediction scheme to interpolate missing control data. Coupled with a transmit power control and a reliable delivery mechanisms *Jerk Beaconing*, while using on average only 25 % of the resources used by the static approach, improves the overall safety of the system. This result has shown that we can do better than sticking to the commonly assumed 10 Hz rule even for an automated control system.

Finally, we proposed a Cooperative Adaptive Cruise Control (CACC) algorithm where, as opposed to the approaches in the literature, the communication topology is configurable. The benefit is that we can potentially adapt the communication pattern to the characteristics of the network. We thoroughly analyzed the control algorithm in very challenging scenarios, including realistic network disturbances and detailed engine modeling, showing its robustness, stability, and convergence characteristics.

As mentioned several times in this thesis, platooning is extremely challenging and fascinating at the same time because of its inter-disciplinary nature. We believe there are still some open questions that should be answered before platooning can become a reality. From a safety perspective, we still need to understand the best action (in terms of safety) in case of an emergency, such as a communication failure. Only a few works tried to deal with fault detection and handling [39, 85, 87] and we need to develop some mechanisms that ensure a certain level of safety. For example, if we loose network connectivity, the system must be able to ensure the safety of the passengers up to some extent. What is the best thing to do? Use a Visible Light Communication (VLC) or a cellular link to perform an emergency split maneuver? Increase the cruising distance between vehicles? Fall back to a sensor based system to split the vehicles and give the control back to the driver? This question is not easy to answer and requires a dedicated in-depth study.

Furthermore, the realistic modeling of vehicle dynamics can still be improved. For example, PLEXE considers air friction force, but it does not consider the reduction of the air drag coefficient when traveling close to a front vehicle [51]. This would enhance simulation realism because acceleration and braking capabilities of a vehicle would also depend on how close to each other vehicles are traveling. Moreover, it would also allow us to measure the reduction of tractive effort, which would give us an idea of the amount of fuel saved.

Another open issue is platooning management. We can find different papers describing strategies and protocols [44, 72, 76, 93] but we need to decide a definite set of strategies to apply. How should we order vehicles in a platoon and based on what? Should we order them at all? What are the maneuvers that vehicles can do? In which conditions should such maneuvers be prohibited? What should we do if a maneuver needs to be aborted? This will need to be coupled with a revenue scheme to re-balance who is driving the platoon (which saves less fuel and has more responsibilities) and who is following.

Besides the technological and communicative aspects we have legislation, which must be changed to account for automated driving. Similarly, driving licenses will need to consider platooning scenarios during training. The problem of real world training is that it might be difficult and expensive to perform tests on the real road because we need multiple vehicles. Interactive simulation can help tackling this issue. We envision an extension of PLEXE where SUMO is replaced by a real time 3D simulator that can be used to train future drivers. This would allow them to get a basic understanding, familiarize with the interface, learn how to safely perform maneuvers, and how to manage emergency situations. Such a tool would also be extremely useful for the scientific community, as this would enable to develop and test different user interfaces, which should be standardized among all manufacturers.

Finally, a fundamental issue is privacy because vehicles will need to share data. For example, to decide the ordering of vehicles in a platoon they might need to share their highway exit, and this might be undesirable. The identity of the driver or the vehicle might also be needed for the revenue system to work or, for example, for insurance reasons. Can we find a way to share this information while preserving privacy and avoid tracking of the drivers?

Even if the idea of platooning is under active development since more than two decades, some questions are still open. We however believe that the scientific community will soon answer them, finally bringing platooning to life: This will completely change how we conceive driving.

List of Abbreviations

ABS	Antilock Braking System
AC	Access Category
ACC	Adaptive Cruise Control
ADAS	Advanced Driving Assistance Systems
BPSK	Binary Phase-Shift Keying
BSM	Basic Safety Message
BSS	Basic Service Set
CACC	Cooperative Adaptive Cruise Control
CAM	Cooperative Awareness Message
CC	Cruise Control
CCA	Clear Channel Assessment
CCH	Control Channel
CSMA/CA	Carrier Sensing Multiple Access with Collision Avoidance
DCC	Decentralized Congestion Control
DENM	Decentralized Environmental Notification Message
D-FPAV	Distributed Fair Power Adjustment for Vehicular environments
DP	DCC Profile
DSC	DCC Sensitivity Control
DSRC	Dedicated Short Range Communications
DynB	Dynamic Beaconing
ECDF	Empirical Cumulative Density Function
EEBL	Emergency Electronic Brake Light
EMBARC	Error Model based Adaptive Rate Control
ETSI	European Telecommunications Standards Institute
EU	European Union
FCC	Federal Communications Commission

FOT	Field Operational Test
GDP	Gross Domestic Product
GI	Guard Interval
GPS	Global Positioning System
ICWS	Intersection Collision Warning System
IDM	Intelligent Driver Model
IEEE	Institute of Electrical and Electronics Engineers
ISI	Inter Symbol Interference
ISM	Industrial, Scientific and Medical
ITS	Intelligent Transportation System
IVC	Inter-Vehicle Communication
LIMERIC	LInear MEssage Rate Integrated Control
LOS	Line Of Sight
LTE	Long Term Evolution
MAC	Medium Access Control
MSDU	MAC Service Data Unit
NFC	Near Field Communication
NHTSA	National Highway Traffic Safety Administration
NIC	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
PHY	Physical Layer
PI	Proportional Integral
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
SCH	Service Channel
STE	Suspected Tracking Error
SUMO	Simulation of Urban MObility
TDC	Transmit Datarate Control
TDMA	Time Division Multiple Access
TIS	Traffic Information System
TPC	Transmit Power Control
TRC	Transmit Rate Control
UMTS	Universal Mobile Telecommunications System
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad Hoc Network
VLC	Visible Light Communication

WAVE	Wireless Access in Vehicular Environments

List of Figures

1.1 Statistics on the total number of cars per country and the number of cars per inhabitant. Sources EUROSTAT [5], U.S. Departments of Transportation [6], and U.S. Census Bureau [22]. 2

1.2 Fatalities count and fatalities split by road type for year 2011. Sources European Commission [7] and National Highway Traffic Safety Administration (NHTSA) [4]. 3

1.3 Traffic flow improvement by increasing maximum density for a free-flow speed $v_m = 100\,\text{km/h}$. 4

1.4 Maximum road utilization as function of speed when respecting safety distance. 6

1.5 Air drag for a single car and two cars traveling close to each other. 6

2.1 Car accidents reduction thanks to the Emergency Electronic Brake Light (EEBL) application, as a function of the market penetration rate [101]. Copyright © 2013 IEEE. 14

2.2 Decentralized Congestion Control (DCC) state machines. 20

2.3 A simple feedback control system. 24

2.4 First order lag applied to control input u for $\tau = 0.5\,\text{s}$. 24

2.5 Artificially generated inter-vehicle distance traces showing string stable and string unstable behavior. 26

2.6 Comparison of a proportional and a proportional integral Cruise Control (CC) subject to an external disturbance. In this plot, $\dot{x}_d = 30\,\text{m/s}$, $\tau = 0.5\,\text{s}$, $k_p = 1$, and $k_i = 0.5$ (for the Proportional Integral (PI) controller). 27

2.7 Speed profiles for an 8-car platoon using ACC and PATH's CACC [99]. Copyright © 2014 IEEE. 30

2.8 Maximum acceleration curves as a function of vehicle speed and parameterized with respect to the different gears in the real case of an Audi R8 4.2 FSI Quattro car (2007). 34

3.1 Schematic structure of the simulator. Copyright © 2014 IEEE. 41
3.2 Polynomials of different degrees fitting measured power curve. The black dots represent the measured points, the red curve the fitting polynomial, and the orange line is the corresponding torque curve. 44
3.3 State machines of the sample join maneuver. Copyright © 2014 IEEE. . . . 47
3.4 Vehicles speed for the different implemented cruise controllers showing string-stability properties. Copyright © 2014 IEEE. 49
3.5 Vehicle dynamics for the join maneuver. The left column shows the results for $\xi = 1$, $\omega_n = 0.2\,\mathrm{Hz}$, while the right column for $\xi = 2$, $\omega_n = 1\,\mathrm{Hz}$. Copyright © 2014 IEEE. 50
3.6 Graphical sketch of different situations for the join-at-middle maneuver. Automated cars shown in dark color. Copyright © 2014 IEEE. 51
3.7 Vehicles dynamics for scenarios 0 and 1: left plots show the GPS distance to the leader; right plots show the measured radar distance. Copyright © 2014 IEEE. 54
3.8 Vehicles dynamics for scenarios 2 and 3: left plots show the GPS distance to the leader; right plots show the measured radar distance. Copyright © 2014 IEEE. 55
3.9 Vehicles acceleration and speed profiles when using the realistic engine model. 56

4.1 Cars used for the experimental validation. Copyright © 2014 IEEE. 61
4.2 Comparison of received power distributions between the experimental testbed (measurement) and the simulation environment (model), for the communication between immediate followers (0 being the leader, 3 being the last car). Transmit power 20 dBm for the leader and 0 dBm for the followers. 1 → 2 and 2 → 3 model curves overlap because the distance and the transmit power between vehicles 1-2 and 2-3 are the same. Copyright © 2015 IEEE. 62
4.3 Busy time ratio and collisions for the 160 and the 640 cars scenarios, CCA-threshold set to −95 dBm. Copyright © 2015 IEEE. 65
4.4 Safe time ratios of both leader and front messages for the 160 and the 640 cars scenarios, CCA-threshold set to −95 dBm. Copyright © 2015 IEEE. . . 67

4.5 Safe time ratio of leader messages for the slotted approaches, 160 cars scenario, for different values of CCA-threshold. Copyright © 2015 IEEE. . . 69

4.6 Safe time ratio of leader messages for the slotted approach with transmit power control, 640 cars scenario, for different values of CCA-threshold. Copyright © 2015 IEEE. 69

4.7 Safe time ratio of leader and front messages for STBP and SLBP, for the scenario with human driven vehicles. CCA-threshold set to −65 dBm. Copyright © 2015 IEEE. 70

4.8 Minimum distances after the complete stop of the platoon as function of the beacon interval, for different leader decelerations. Copyright © 2015 IEEE. 71

4.9 Δ_{msg} function for $\max_{\mathrm{bi}} = 1\,\mathrm{s}$, $\min_{\mathrm{bi}} = 0.1\,\mathrm{s}$, $\Delta_{u_{\max}} = 2\,\mathrm{m/s}^2$, and different values of p (0.1, 0.3, 0.5, 1, 3 and 100). Copyright © 2015 IEEE. 73

4.10 Flow charts for the main protocol loop, sending, and ack timeout procedures. Copyright © 2015 IEEE. 75

4.11 Flow charts for message reception handling. Copyright © 2015 IEEE. 76

4.12 Minimum distances and channel busy ratio for the different protocols when transmit power control is not employed, harsh scenario. Copyright © 2015 IEEE. 79

4.13 Minimum distances for the different protocols, 640 cars. Copyright © 2015 IEEE. 80

4.14 Channel busy time for the different protocols, 640 cars. Copyright © 2015 IEEE. 81

4.15 Empirical CDF of beacon inter-arrival times for the different protocols, 640 cars. Copyright © 2015 IEEE. 81

5.1 Screenshot of the realistic scenario. Human-driven vehicles in white, blue, and yellow, and platooning cars in red on the left-most lane. Copyright © 2015 IEEE. 91

5.2 Control topology selected for the simulations. Copyright © 2015 IEEE. . . . 92

5.3 Basic convergence analysis with $v_0 = 100\,\mathrm{km/h}$, $N+1 = 8$ vehicles. Platoon creation and maintenance. Copyright © 2015 IEEE. 92

5.4 Leader tracking maneuver: Time history of vehicle speeds. Copyright © 2015 IEEE. 94

5.5 Braking maneuver: Time history of bumper to bumper distances. Copyright © 2015 IEEE. 94

5.6 Robustness with respect to a sinusoidal disturbance: Time history of bumper to bumper distances. Copyright © 2015 IEEE. 94
5.7 Dynamics for a platoon of $N + 1 = 16$ vehicles. Copyright © 2015 IEEE. . . 95
5.8 Leader tracking maneuver in the realistic network scenario: Time history of vehicle speeds. Copyright © 2015 IEEE. 95
5.9 Sinusoidal disturbance on leader motion in the realistic network scenario: Time history of bumper to bumper distances. Copyright © 2015 IEEE. . . . 96
5.10 Platoon creation and maintaining: Convergence analysis with realistic vehicles models ($N + 1 = 8$ and $v_0 = 100\,\mathrm{km/h}$). 96
5.11 Leader tracking maneuver with realistic vehicle model. 97
5.12 Leader tracking maneuver with realistic network scenario and vehicle model. Time history of vehicle speeds. 97
5.13 Sinusoidal disturbance acting on leader motion with realistic network scenario and vehicle model. Time history of bumper to bumper distances. . . . 98

List of Tables

2.1 DCC parameters for the control channel, AC_VI. Copyright © 2015 IEEE. . 21

3.1 Network and road traffic simulation parameters. Copyright © 2014 IEEE. . 47

4.1 Parameters employed in the experimental validation. Copyright © 2015 IEEE. 61
4.2 Network and road traffic simulation parameters. Copyright © 2015 IEEE. . 63
4.3 Network and road traffic simulation parameters. Copyright © 2015 IEEE. . 78

5.1 Network simulation parameters. 91
5.2 Traffic simulation parameters for the realistic scenario. Copyright © 2015 IEEE. 91

Bibliography

[1] "Wireless Access in Vehicular Environments," IEEE, Std 802.11p-2010, July 2010.

[2] "Roadmap to a Single European Transport Area - Towards a Competitive and Resource Efficient Transport System," European Commission, Tech. Rep., March 2011.

[3] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Std 802.11-2012, 2012.

[4] "2013 Motor Vehicle Crashes: Overview," National Center for Statistics and Analysis, National Highway Traffic Safety Administration, Traffic Safety Facts Research Note DOT HS 812 101, December 2014.

[5] "Energy, Transport and Environment Indicators," Eurostat, Technical Report KS-DK-14-001, February 2015.

[6] "National Transportation Statistics," U.S. Department of Transportation, Bureau of Transportation Statistics, Tech. Rep., October 2015.

[7] "Traffic Safety Basic Facts on Main Figures," European Commission, Directorate General for Transport, Tech. Rep., June 2015.

[8] S. Abbas and A. Sheikh, "On Understanding the Nature of Slow Fading in LOS Microcellular Channels," in *47th IEEE Vehicular Technology Conference (VTC1997)*. Phoenix, AZ: IEEE, May 1997, pp. 662–666.

[9] M. Abualhoul, M. Marouf, O. Shagdar, and N. Fawzi, "Platooning Control Using Visible Light Communications: A Feasibility Study," in *IEEE Intelligent Transportation Systems Conference (ITSC 2013)*. The Hague, The Netherlands: IEEE, October 2013.

[10] J. Ahn, Y. Wang, B. Yu, F. Bai, and B. Krishnamachari, "RISA: Distributed Road Information Sharing Architecture," in *31st IEEE Conference on Computer Communications (INFOCOM 2012)*. Orlando, FL: IEEE, March 2012, pp. 1494–1502.

[11] A. Ali, G. Garcia, and P. Martinet, "The Flatbed Platoon Towing Model for Safe and Dense Platooning on Highways," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 1, pp. 58–68, January 2015.

[12] G. Bansal, J. Kenney, and C. Rohrs, "LIMERIC: A Linear Adaptive Message Rate Algorithm for DSRC Congestion Control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, November 2013.

[13] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "EMBARC: Error Model based Adaptive Rate Control for Vehicle-to-Vehicle Communications," in *10th ACM International Workshop on Vehicular Internetworking (VANET 2013)*. Taipei, Taiwan: ACM, June 2013, pp. 41–50.

[14] G. Bansal, B. Cheng, A. Rostami, K. Sjoberg, J. B. Kenney, and M. Gruteser, "Comparing LIMERIC and DCC Approaches for VANET Channel Congestion Control," in *6th IEEE International Symposium on Wireless Vehicular Communications (WiVec 2014)*. Vancouver, Canada: IEEE, September 2014, pp. 1–7.

[15] C. Bergenhem, E. Hedin, and D. Skarin, "Vehicle-to-Vehicle Communication for a Platooning System," in *Transport Research Arena*, Athens, Greece, April 2012.

[16] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of Platooning on Public Motorways," in *17th World Congress on Intelligent Transport Systems (ITS 2010)*, Busan, Korea, October 2010.

[17] J. J. Blum and A. Eskandarian, "Avoiding Timeslot Boundary Synchronization for Multihop Message Broadcast in Vehicular Networks," in *69th IEEE Vehicular Technology Conference (VTC2009-Spring)*. Barcelona, Spain: IEEE, April 2009, pp. 1–5.

[18] M. Boban, T. Vinhosa, J. Barros, M. Ferreira, and O. K. Tonguz, "Impact of Vehicles as Obstacles in Vehicular Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 29, no. 1, pp. 15–28, January 2011.

[19] A. Böhm, M. Jonsson, and E. Uhlemann, "Co-Existing Periodic Beaconing and Hazard Warnings in IEEE 802.11p-based Platooning Applications," in *10th ACM*

International Workshop on Vehicular Internetworking (VANET 2013). Taipei, Taiwan: ACM, June 2013, pp. 99–102.

[20] A. Bose and P. Ioannou, "Analysis of Traffic Flow with Mixed Manual and Intelligent Cruise Control Vehicles: Theory and Experiments," University of California, Berkeley, California PATH Research Report UCB-ITS-PRR-2001-13, April 2001.

[21] W. Chen, R. Guha, J. Chennikara-Varghese, M. Pang, R. Vuyyuru, and J. Fukuyama, "Context-driven Disruption Tolerant Networking for Vehicular Applications," in *2nd IEEE Vehicular Networking Conference (VNC 2010)*. Jersey City, NJ: IEEE, December 2010, pp. 33–40.

[22] S. L. Colby and J. M. Ortman, "Projections of the Size and Composition of the U.S. Population: 2014 to 2060," U.S. Census Bureau, Tech. Rep., March 2015.

[23] C. Collin, "Statistics in Focus," Eurostat, Tech. Rep. CA-NZ-00-003, July 2000.

[24] Z. Cui, C. Wang, and H.-M. Tsai, "Characterizing Channel Fading in Vehicular Visible Light Communications with Video Data," in *6th IEEE Vehicular Networking Conference (VNC 2014)*. Paderborn, Germany: IEEE, December 2014, pp. 226–229.

[25] A. Davila and M. Nombela, "Platooning - Safe and Eco-Friendly Mobility," in *SAE 2012 World Congress & Exibition*. Detroit, Michigan: SAE, April 2012.

[26] A. Day, *Braking of Road Vehicles*. Elsevier Science, March 2014.

[27] M. di Bernardo, A. Salvi, and S. Santini, "Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time Varying Heterogeneous Communication Delays," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 102–112, September 2014.

[28] D. Eckhoff and C. Sommer, "A Multi-Channel IEEE 1609.4 and 802.11p EDCA Model for the Veins Framework," in *5th ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2012): 5th ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2012), Poster Session*. Desenzano, Italy: ACM, March 2012.

[29] D. Eckhoff, C. Sommer, and F. Dressler, "On the Necessity of Accurate IEEE 802.11p Models for IVC Protocol Simulation," in *75th IEEE Vehicular Technology Conference (VTC2012-Spring)*. Yokohama, Japan: IEEE, May 2012, pp. 1–5.

[30] L. Elefteriadou, *An Introduction to Traffic Flow Theory*. Springer New York, 2014.

[31] M. Elshenawy, M. El-Darieby, and B. Abdulhai, “Vehicular Mobility Management Schemes for Balancing Loads among WMN Access Points,” in *69th IEEE Vehicular Technology Conference (VTC2009-Spring)*. Barcelona, Spain: IEEE, April 2009, pp. 1–5.

[32] European Telecommunications Standards Institute, “Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part,” ETSI, TS 102 687 V1.1.1, July 2011.

[33] European Telecommunications Standards Institute, “Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band,” ETSI, TS 102 724 V1.1.1, October 2012.

[34] European Telecommunications Standards Institute, “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,” ETSI, TS 302 637-2 V1.3.1, September 2014.

[35] P. Fernandes and U. Nunes, “Platooning of Autonomous Vehicles with Intervehicle Communications in SUMO Traffic Simulator,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC 2010)*, Madeira Island, Portugal, September 2010, pp. 1313–1318.

[36] P. Fernandes and U. Nunes, “Platooning With IVC-Enabled Autonomous Vehicles: Strategies to Mitigate Communication Delays, Improve Safety and Traffic Flow,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 91–106, March 2012.

[37] J. Gerdes and J. Hedrick, “Vehicle Speed and Spacing Control via Coordinated Throttle and Brake Actuation,” *Control Engineering Practice*, vol. 5, no. 11, pp. 1607–1614, 1997.

[38] T. D. Gillespie, *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, 1992, vol. 400.

[39] D. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. Lindsey, “Communication Protocols for a Fault-tolerant Automated Highway System,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 787–800, September 2000.

[40] H. Greenberg, "An Analysis of Traffic Flow," *Operations Research*, vol. 7, no. 1, pp. 79–85, February 1959.

[41] C. Guo, N. Wan, S. Mita, and M. Yang, "Self-defensive Coordinated Maneuvering of an Intelligent Vehicle Platoon in Mixed Traffic," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*. Anchorage, AK: IEEE, September 2012, pp. 1726–1733.

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

[43] S. Hallé and B. Chaib-draa, "A Collaborative Driving System Based on Multiagent Modelling and Simulations," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 4, pp. 320–345, August 2005.

[44] S. Hallé, J. Laumonier, and B. Chaib-Draa, "A Decentralized Approach to Collaborative Driving Coordination," in *IEEE Intelligent Transportation Systems Conference (ITSC 2004)*. Washington, DC: IEEE, October 2004, pp. 453–458.

[45] C. Han, M. Dianati, R. Tafazolli, X. Liu, and X. S. Shen, "A Novel Distributed Asynchronous Multichannel MAC Scheme for Large-Scale Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3125–3138, September 2012.

[46] H. Hao and P. Barooah, "Stability and Robustness of Large Platoons of Vehicles with Double-integrator Models and Nearest Neighbor Interaction," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 18, pp. 2097–2122, December 2013.

[47] H. Hartenstein and K. Laberteaux, Eds., *VANET - Vehicular Applications and Inter-Networking Technologies*, ser. Intelligent Transport Systems. John Wiley & Sons, December 2010.

[48] C.-L. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Adaptive Intervehicle Communication Control for Cooperative Safety Systems," *IEEE Network*, vol. 24, no. 1, pp. 6–13, January 2010.

[49] C.-L. Huang, X. Guan, Y. Fallah, R. Sengupta, and H. Krishnan, "Robustness Evaluation of Decentralized Self-Information Dissemination Control Algorithms for VANET Tracking Applications," in *70th IEEE Vehicular Technology Conference (VTC2009-Fall)*. Anchorage, AK: IEEE, September 2009, pp. 1–5.

[50] M. Huang and J. H. Manton, “Coordination and Consensus of Networked Agents with Noisy Measurements: Stochastic Algorithms and Asymptotic Behavior,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 134–161, February 2009.

[51] W.-H. Hucho, *Aerodynamics of Road Vehicles: From Fluid Mechanics to Vehicle Engineering*. Elsevier, 1987.

[52] IEEE, “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager,” IEEE, Std 1609.1-2006, October 2006.

[53] IEEE, “IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services,” IEEE, Std 1609.3-2010, December 2010.

[54] IEEE, “IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation,” IEEE, Std 1609.4-2010, February 2011.

[55] IEEE, “IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Over-the-Air Electronic Payment Data Exchange Protocol for Intelligent Transportation Systems (ITS),” IEEE, Std 1609.11-2010, January 2011.

[56] IEEE, “IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Identifier Allocations,” IEEE, Std 1609.12-2012, September 2012.

[57] IEEE, “IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages,” IEEE, Std 1609.2-2013, April 2013.

[58] IEEE, “IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture,” IEEE, Std 1609.0-2013, March 2014.

[59] D. Jia, K. Lu, and J. Wang, “A Disturbance-adaptive Design for VANET-enabled Vehicle Platoon,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 527–539, February 2014.

[60] D. Jiang, Q. Chen, and L. Delgrossi, “Optimal Data Rate Selection for Vehicle Safety Communications,” in *5th ACM International Workshop on Vehicular Inter-Networking (VANET 2008)*, San Francisco, CA, September 2008, pp. 30–38.

[61] D. Jiang and L. Delgrossi, “IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments,” in *67th IEEE Vehicular Technology Conference (VTC2008-Spring)*, Marina Bay, Singapore, May 2008, pp. 2036–2040.

[62] S. Joerer, B. Bloessl, M. Segata, C. Sommer, R. Lo Cigno, and F. Dressler, “Fairness Kills Safety: A Comparative Study for Intersection Assistance Applications,” in *25th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2014)*. Washington, D.C.: IEEE, September 2014, pp. 1442–1447.

[63] S. Joerer, F. Dressler, and C. Sommer, “Comparing Apples and Oranges? Trends in IVC Simulations,” in *9th ACM International Workshop on Vehicular Internetworking (VANET 2012)*. Low Wood Bay, Lake District, UK: ACM, June 2012, pp. 27–32.

[64] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, “To Crash or Not to Crash: Estimating its Likelihood and Potentials of Beacon-based IVC Systems,” in *4th IEEE Vehicular Networking Conference (VNC 2012)*. Seoul, Korea: IEEE, November 2012, pp. 25–32.

[65] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, “A Vehicular Networking Perspective on Estimating Vehicle Collision Probability at Intersections,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1802–1812, May 2014.

[66] P. S. Jootel, “SAfe Road TRains for the Environment,” SARTRE Project, Final Project Report, October 2012.

[67] J. B. Kenney, G. Bansal, and C. E. Rohrs, “LIMERIC: a Linear Message Rate Control Algorithm for Vehicular DSRC Systems,” in *8th ACM International Workshop on Vehicular Internetworking (VANET 2011)*. Las Vegas, NV: ACM, September 2011, pp. 21–30.

[68] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline, and Vehicle*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2005.

[69] D. Krajzewicz, R. Blokpoel, F. Cartolano, P. Cataldi, A. Gonzalez, O. Lazaro, J. Leguay, L. Lin, J. Maneros, and M. Rondinone, “iTETRIS - A System for the Evaluation of Cooperative Traffic Management Solutions,” in *14th International Forum on Advanced Microsystems for Automotive Applications (AMAA 2010)*, Berlin, Germany, May 2010.

[70] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “SUMO (Simulation of Urban MObility); An Open-source Traffic Simulation,” in *4th Middle East Symposium on Simulation and Modelling (MESM 2002)*, Sharjah, United Arab Emirates, September 2002, pp. 183–187.

[71] S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," PhD Thesis, University of Cologne, 1998.

[72] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*. Wollogong, Australia: IEEE, July 2013, pp. 1152–1157.

[73] C. Lei, E. van Eenennaam, W. Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg, "Impact of Packet Loss on CACC String Stability Performance," in *11th International Conference on ITS Telecommunications (ITST 2011)*, Saint Petersburg, Russia, August 2011, pp. 381–386.

[74] T. Li and J.-F. Zhang, "Consensus Conditions of Multi-agent Systems with Time-varying Topologies and Stochastic Communication Noises," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2043–2057, September 2010.

[75] X. Liu, A. Goldsmith, S. Mahal, and J. Hedrick, "Effects of Communication Delay on String Stability in Vehicle Platoons," in *IEEE Intelligent Transportation Systems Conference (ITSC 2001)*. Oakland, CA: IEEE, August 2001, pp. 625–630.

[76] F. Michaud, P. Lepage, P. Frenette, D. Letourneau, and N. Gaubert, "Coordinated Maneuvering of Automated Vehicles in Platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 437–447, December 2006.

[77] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative Adaptive Cruise Control in Real Traffic Situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, February 2014.

[78] T. Murray, M. Cojocari, and H. Fu, "Measuring the Performance of IEEE 802.11p using NS-2 Simulator for Vehicular Networks," in *IEEE International Conference on Electro/Information Technology (EIT 2008)*. Ames, IA: IEEE, May 2008, pp. 498–503.

[79] National Highway Traffic Safety Administration, "Vehicle Safety Communications Project Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled by DSRC," NHTSA, Project Report DOT HS 809 859, March 2005.

[80] G. J. Naus, R. P. Vugts, J. Ploeg, M. Van de Molengraft, and M. Steinbuch, "String-stable CACC Design and Experimental Validation: A Frequency-domain Approach,"

IEEE Transactions on Vehicular Technology, vol. 59, no. 9, pp. 4268–4279, November 2010.

[81] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[82] S. Oncu, J. Ploeg, N. Van De Wouw, and H. Nijmeijer, "Cooperative Adaptive Cruise Control: Network-Aware Analysis of String Stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, August 2014.

[83] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Joint Traffic and Network Simulator," in *13th ACM International Conference on Mobile Computing and Networking (MobiCom 2007)*, Montreal, Canada, September 2007.

[84] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control," in *IEEE International Conference on Intelligent Transportation Systems (ITSC 2011)*. Washington, DC: IEEE, October 2011, pp. 260–265.

[85] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of CACC Performance Subject to Unreliable Wireless Communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. The Hague, The Netherlands: IEEE, October 2013, pp. 1210–1216.

[86] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer, 2012.

[87] R. Rajamani, A. Howell, C. Chen, J. Hedrick, and M. Tomizuka, "A Complete Fault Diagnostic System for Automated Vehicles Operating in a Platoon," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 4, pp. 553–564, July 2001.

[88] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, July 2000.

[89] K. Santhanakrishnan and R. Rajamani, "On Spacing Policies for Highway Vehicle Automation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 198–204, December 2003.

[90] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno, "A Consensus-based Approach for Platooning with Inter-Vehicular Communications," in *34th IEEE Conference on Computer Communications (INFOCOM 2015)*. Hong Kong: IEEE, April 2015, pp. 1158–1166.

[91] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno, "A Consensus-based Approach for Platooning with Inter-Vehicular Communications and its Validation in Realistic Scenarios," in *IEEE Transactions on Vehicular Technology*. IEEE, 2015, submitted for peer review.

[92] R. K. Schmidt, A. Brakemeier, T. Leinmüller, F. Kargl, and G. Schäfer, "Advanced Carrier Sensing to Resolve Local Channel Congestion," in *8th ACM International Workshop on Vehicular Internetworking (VANET 2011)*. Las Vegas, Nevada: ACM, September 2011, pp. 11–20.

[93] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. Lo Cigno, "Supporting Platooning Maneuvers through IVC: An Initial Protocol Analysis for the Join Maneuver," in *11th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2014)*. Obergurgl, Austria: IEEE, April 2014, pp. 130–137.

[94] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards Inter-Vehicle Communication Strategies for Platooning Support," in *7th IFIP/IEEE International Workshop on Communication Technologies for Vehicles (Nets4Cars 2014-Fall)*. Saint-Petersburg, Russia: IEEE, October 2014, pp. 1–6.

[95] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards Communication Strategies for Platooning: Simulative and Experimental Evaluation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5411–5423, December 2015.

[96] M. Segata, B. Bloessl, S. Joerer, C. Sommer, R. Lo Cigno, and F. Dressler, "Vehicle Shadowing Distribution Depends on Vehicle Type: Results of an Experimental Study," in *5th IEEE Vehicular Networking Conference (VNC 2013)*. Boston, MA: IEEE, December 2013, pp. 242–245.

[97] M. Segata, F. Dressler, and R. Lo Cigno, "Jerk Beaconing: A Dynamic Approach to Platooning," in *7th IEEE Vehicular Networking Conference (VNC 2015)*. Kyoto, Japan: IEEE, December 2015, pp. 135–142.

[98] M. Segata, F. Dressler, R. Lo Cigno, and M. Gerla, “A Simulation Tool for Automated Platooning in Mixed Highway Scenarios,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 4, pp. 46–49, October 2012.

[99] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, “PLEXE: A Platooning Extension for Veins,” in *6th IEEE Vehicular Networking Conference (VNC 2014)*. Paderborn, Germany: IEEE, December 2014, pp. 53–60.

[100] M. Segata and R. Lo Cigno, “Emergency Braking: a Study of Network and Application Performance,” in *8th ACM International Workshop on Vehicular Internetworking (VANET 2011)*. Las Vegas, NV: ACM, September 2011, pp. 1–10.

[101] M. Segata and R. Lo Cigno, “Automatic Emergency Braking - Realistic Analysis of Car Dynamics and Network Performance,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4150–4161, October 2013.

[102] M. Segata, R. Lo Cigno, H.-M. Tsai, and F. Dressler, “On Platooning Control using IEEE 802.11p in Conjunction with Visible Light Communications,” in *12th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2016)*. Cortina d’Ampezzo, Italy: IEEE, January 2016, to appear.

[103] M. Sepulcre, J. Gozalvez, O. Altintas, and H. Kremo, “Adaptive Beaconing for Congestion and Awareness Control in Vehicular Networks,” in *Vehicular Networking Conference (VNC), 2014 IEEE*. Paderborn, Germany: IEEE, December 2014, pp. 81–88.

[104] S. Shladover, “PATH at 20 – History and Major Milestones,” in *IEEE Intelligent Transportation Systems Conference (ITSC 2006)*, Toronto, Canada, September 2006, pp. 22–29.

[105] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, November 2014.

[106] C. Sommer, D. Eckhoff, R. German, and F. Dressler, “A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments,” in *8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011)*. Bardonecchia, Italy: IEEE, January 2011, pp. 84–90.

[107] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[108] C. Sommer, S. Joerer, and F. Dressler, "On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation," in *4th IEEE Vehicular Networking Conference (VNC 2012)*. Seoul, Korea: IEEE, November 2012, pp. 64–69.

[109] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," in *32nd IEEE Conference on Computer Communications (INFOCOM 2013), Mini-Conference*. Turin, Italy: IEEE, April 2013, pp. 110–114.

[110] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. Lo Cigno, and F. Dressler, "How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, July 2015.

[111] D. Swaroop and J. Hedrick, "String Stability of Interconnected Systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 349–357, March 1996.

[112] R. Szalai and G. Orosz, "Decomposing the Dynamics of Heterogeneous Delayed Networks with Applications to Connected Vehicle Systems," *Physical Review E*, vol. 88, no. 4, p. 040902, October 2013.

[113] T. Tielert, D. Jiang, H. Hartenstein, and L. Delgrossi, "Joint Power/Rate Congestion Control Optimizing Packet Reception in Vehicle Safety Communications," in *10th ACM International Workshop on Vehicular Internetworking (VANET 2013)*. Taipei, Taiwan: ACM, June 2013, pp. 51–60.

[114] B. Tomas, H.-M. Tsai, and M. Boban, "Simulating Vehicular Visible Light Communication: Physical Radio and MAC Modeling," in *6th IEEE Vehicular Networking Conference (VNC 2014)*. Paderborn, Germany: IEEE, December 2014, pp. 222–225.

[115] O. Tonguz, N. Wisitpongphan, J. Parikh, F. Bai, P. Mudalige, and V. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Networks," in *3rd International Conference on Broadband Communications, Networks, and Systems (BROADNETS)*, San Jose, CA, October 2006.

[116] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3684–3703, September 2009.

[117] M. Treiber, A. Hennecke, and D. Helbing, “Congested Traffic States in Empirical Observations and Microscopic Simulations,” *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, August 2000.

[118] V. Turri, “Fuel-efficient and Safe Heavy-duty Vehicle Platooning Through Look-ahead Control,” PhD Thesis, KTH Royal Institute of Technology, October 2015.

[119] B. van Arem, C. van Driel, and R. Visser, “The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, December 2006.

[120] A. Varga, “The OMNeT++ Discrete Event Simulation System,” in *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, June 2001.

[121] L. Y. Wang, A. Syed, G. G. Yin, A. Pandya, and H. Zhang, “Control of Vehicle Platoons for Highway Safety and Efficient Utility: Consensus with Communications and Vehicle Dynamics,” *Journal of Systems Science and Complexity*, vol. 27, no. 4, pp. 605–631, August 2014.

[122] S.-Y. Wang and C.-C. Lin, “NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches,” in *68th IEEE Vehicular Technology Conference (VTC2008-Fall)*, Calgary, Canada, September 2008, pp. 1–2.

[123] N. Wisitpongphan, F. Bai, P. Mudalige, and O. K. Tonguz, “On the Routing Problem in Disconnected Vehicular Ad-hoc Networks,” in *26th IEEE Conference on Computer Communications (INFOCOM 2007)*, Anchorage, AK, May 2007, pp. 2291–2295.

[124] J. Y. Wong, *Theory of Ground Vehicles*, 3rd ed. John Wiley & Sons, 2001.

[125] G. Yin, Y. Sun, and L. Y. Wang, “Asymptotic Properties of Consensus-type Algorithms for Networked Systems with Regime-switching Topologies,” *Automatica*, vol. 47, no. 7, pp. 1366–1378, July 2011.

[126] S. Yousefi, T. Chahed, S. M. M. Langari, and K. Zayer, “Comfort Applications in Vehicular Ad Hoc Networks Based on Fountain Coding,” in *71st IEEE Vehicular Technology Conference (VTC2010-Spring)*. Taipei, Taiwan: IEEE, May 2010, pp. 1–5.

[127] S. Zemouri, S. Djahel, and J. Murphy, “Smart Adaptation of Beacons Transmission Rate and Power for Enhanced Vehicular Awareness in VANETs,” in *17th IEEE International Conference on Intelligent Transportation Systems (ITSC 2014)*. Qingdao, China: IEEE, October 2014, pp. 739–746.

[128] L. Zhang and S. Valaee, “Safety Context-aware Congestion Control for Vehicular Broadcast Networks,” in *15th IEEE International Symposium on Signal Processing Advances in Wireless Communications (SPAWC 2014)*. Toronto, Canada: IEEE, June 2014, pp. 399–403.

[129] L. Zhao and J. Sun, “Simulation Framework for Vehicle Platooning and Car-following Behaviors Under Connected-vehicle Environment,” *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 914–924, November 2013.

[130] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, “Study on Emergency-avoidance Braking for the Automatic Platooning of Trucks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1748–1757, August 2014.

Acknowledgements

The feeling is kind of strange. I can clearly remember Renato (at that time he was my master thesis' advisor) asking me: "What about doing a PhD?". My answer was clear: "No way! Never in my life!". And indeed here we are. I have to admit that these four years had been wonderful. One might think that this is because I traveled around the world payed by the government, but that is only a minor part. During this experience I learned a lot, not only from a scientific perspective. I learned tools, I learned methodologies, I got to know so many people and different perspectives. In the end, this thesis is not only the result of my PhD work, but a mix of all the things I learned. This would not have been possible without the people that have been part of this experience, and that are still part of my current (and hopefully, my future) life.

I will obviously start with her, Caroline. She is the one that is always there for me, the one that kept me motivated and cheered me up during bad and stressful periods of this PhD. With her I shared wonderful moments: I was lucky enough to meet someone that woke up my passion for mountaineering, and that shares that passion with me. But it is not only about this. Even simply staying at home watching a movie or a TV series is extremely relaxing. Or even if she is watching a movie seating next to me in the evening while I am still working makes everything easier. She had been very patient during these years, and I will never be thankful enough. To just give you an idea, I can tell you about when we moved together. It was in 2012, the 1st of April. April fools' day, you might think, and indeed it looked somehow like that, because on the 2nd of April I moved to Innsbruck (alone) to start my PhD. Every normal person would have gone crazy, but she didn't. She supported me the whole time, making everything easier. I would also like to mention the new member of the family who is due to come in a few weeks. It is very emotional seeing him/her grow (in the baby bump obviously) and I am looking forward to know him/her. The idea of having a new person in our family made me much happier in the latest eight months. Thanks to both of you.

I then have to say something about Falko and Renato. Actually, I don't "have to", but I "want to". They are my PhD advisors and I am referring to them informally using their first names, because in these years they became more like two friends, rather than professors or bosses. If you know "PhD Comics" you know that advisors are usually depicted as horrible persons, and you might think my doctorate must have been twice as terrible. This has never been the case. I found two advisors that shared with me not only their academic and scientific interests, but also their passion for sports. Working and "hanging out" with them has always been fun, and it helped keeping the stress under control. I learned a lot from them during my PhD and I hope to continue working together in the next years.

If I came to this point of my life it is also thanks to my family. Mum and dad always supported me during studies. Thanks to their effort I could always focus on studying without too much worries, obtaining pretty satisfying results. Their support still continue today: For example, my mum is trying to teach me some of her delicious recipes, sometimes trying to trick me ;). The outcome is obviously not as good, but still enough for feeding a family. But there are not only mum and dad around me: There's my brother Alessandro, his wife Francesca, and my niece Arianna, which says she likes math and is already very good at playing Minecraft. A very promising little girl :). Being a little older than me (don't get offended), Ale and Franci are always a point of reference, especially concerning life experience. Soon, they will also be a point of reference for what concerns child education :). And don't forget my parents-in-law Pepi and Barbara, which are always there to help us. Together with Caroline and my father, they introduced me to the majority of mountain sports I currently practice, and hanging out with them is always a lot of fun. As with PhD advisors, some people have terrible experiences with parents-in-law, but again this is not my case :). Sometimes they bring me to my limits when doing sports, but I think it is a plus: It's all training! May I also mention Simon, a great friend that taught me the remainder of the mountain sports I practice. He is playing the role of my personal climbing instructor. He is always extremely patient with me, telling me when I am doing good and, most of the times, when I am doing wrong ("Last week you climbed like s**t!"). Every time he pushes me to the limit, making me stronger and stronger. If I became a fairly decent climber, it is mostly thanks to him. We both like adrenalin, so we always have a lot of fun together. There are obviously many more people in my family: So many that it is impossible to mention all of them. Still, thank to all of you.

Another thing to remember is that you can have a pleasant working environment only if you have good colleagues. As I started in Innsbruck I immediately felt at home. We had a wonderful group that made my working day easier and my spare time wonderful. Leaving Innsbruck was though because it had been one of the best periods of my life, but luckily I got great colleagues in Trento as well, and thanks to them my “handover” had been smooth. Office 141 is a happy and funny place to be. Getting to work every day is easier knowing that you share your office with such great people, even if your alarm rings at 6:15 in the morning.

Last but absolutely not least, there are my friends, which are on my side since I was in high school. We shared so many moments together it is impossible to remember all of them. We have a wonderful group of people with so many different interests. Hanging out is always a lot of fun, and if you have any problem they are always there for you. I am happy that the group is still growing and many people are joining us. It is never boring with you guys. Thanks for being such an amazing group of friends.

There are so many people I missed in this list and that I cannot mention without using another twenty pages. Among them there are university colleagues, university staff, my Innsbruck flatmates, people I got to know during conferences, all the new friends I made in Val Gardena, and many more. With them I enjoyed beautiful moments of my life that I will never forget.

Thanks to all of you.

Ringraziamenti

È una sensazione strana. Ricordo chiaramente Renato (a quei tempi era il mio relatore di tesi magistrale) chiedermi: “Che ne dici di un PhD?”. La mia risposta era stata chiara “Manco per idea!”. Ed infatti eccoci qui. Devo ammettere che questi quattro anni sono stati magnifici. Potreste pensare che sia solo perché ho girato il mondo a spese del governo, ma questa è solo una minima parte della storia. Durante questa esperienza ho imparato un sacco di cose e non solo dal punto di vista scientifico. Ho imparato strumenti, metodologie e diverse prospettive, e ho conosciuto un sacco di persone. A conti fatti, questa tesi non è solo il risultato del lavoro svolto durante il dottorato, ma l’insieme di tutte le cose che ho imparato. Questo non sarebbe stato possibile senza le persone che hanno fatto parte di questa esperienza e che sono ancora parte della mia vita corrente (e futura, si spera).

Ovviamente inizio da lei, Caroline. Lei è la persona che è sempre lì per me, la persona che mi ha motivato e mi ha risollevato l’umore durante i periodi di stress di questo dottorato. Con lei ho passato dei momenti fantastici: sono stato così fortunato da incontrare una persona che ha risvegliato la mia passione per la montagna e che la condivide con me. Ma questo non è tutto. Il semplice stare a casa con lei guardando un film o una serie TV è incredibilmente rilassante. Anche quando lei guarda la TV seduta accanto a me mentre lavoro la sera rende tutto più semplice. Lei è stata molto paziente durante questi anni e non le sarò mai grato abbastanza. Per darvi un’idea, vi posso raccontare di quando siamo andati a vivere insieme. Era il primo aprile 2012. Pesce d’aprile, penserete, e in effetti è andata all’incirca così perché il 2 di aprile mi sono trasferito a Innsbruck (da solo) per iniziare il mio dottorato. Qualunque persona avrebbe dato in escandescenza, ma non lei. Mi ha sempre dato supporto, rendendo tutto più facile. Vorrei anche dire due parole sul nuovo componente della famiglia in arrivo fra qualche settimana. È stato molto emozionante vederlo/a crescere (nel pancione ovviamente) e non vedo l’ora di conoscerlo/a. L’idea di una nuova persona nella nostra famiglia mi ha reso molto più felice in questi ultimi otto mesi. Grazie a entrambi.

Poi devo dire qualcosa di Falko e Renato. In realtà non è che devo farlo, ma voglio farlo. Loro sono i miei relatori ai quali faccio riferimento in maniera informale usando i loro nomi, poiché in questi anni sono diventati degli amici, più che dei professori o dei capi. Se conoscete "PhD Comics" sapete che i relatori sono spesso dipinti come delle persone orribili e potreste pensare che il mio dottorato sia stato il doppio più terribile. Non è mai stato così. Ho trovato due relatori che hanno condiviso con me non solo i loro interessi accademici e scientifici, ma anche la loro passione per lo sport. Lavorare e passare del tempo libero con loro è sempre stato molto divertente, e mi ha aiutato a mantenere sotto controllo lo stress. Ho imparato un sacco di cose da loro e spero di continuare a lavorare insieme nei prossimi anni.

Se sono giunto a questo punto è anche grazie alla mia famiglia. Mamma e papà mi hanno sempre dato supporto durante i miei studi. Grazie ai loro sforzi ho sempre potuto concentrarmi sullo studio senza troppe preoccupazioni, ottenendo risultati soddisfacenti. Il loro supporto continua ancora: mamma, ad esempio, sta cercando di insegnarmi alcune delle sue deliziose ricette, a volte cercando di imbrogliarmi ;). Il risultato non è buono come quando è lei a cucinare, ma abbastanza decente per poter sfamare una famiglia. Ma non ci sono solo mamma e papà ovviamente: ci sono mio fratello Alessandro, sua moglie Francesca e mia nipote Arianna, la quale dice di apprezzare la matematica ed è già molto brava a giocare a Minecraft. Una bambina molto promettente :). Essendo un pelo più vecchi di me (non offendetevi), Ale e Franci sono sempre un punto di riferimento, specialmente per quanto riguarda le esperienze di vita. Presto diventeranno un punto di riferimento anche per quanto riguarda l'educazione dei figli :). E non dimenticate i miei suoceri Pepi e Barbara che sono sempre disponibili ad aiutarci. Assieme a Caroline e a mio papà, mi hanno insegnato la maggioranza degli sport di montagna che pratico, e passare del tempo con loro è sempre divertentissimo. Così come per i relatori di dottorato, certe persone hanno delle terribili esperienze con i propri suoceri, ma di nuovo questo non è il mio caso :). Ogni tanto mi portano al limite quando facciamo sport, ma penso sia un vantaggio: è tutto allenamento! Permettetemi anche di menzionare Simon, un grande amico che mi ha insegnato il resto degli sport di montagna che pratico. Lui ricopre il ruolo di mio istruttore di arrampicata personale. È sempre molto paziente con me, dicendomi cosa faccio di giusto e, la maggior parte delle volte, cosa faccio di sbagliato ("L'ultim ena jives da c***o!"). Cerca sempre di farmi spingere fino al limite facendomi diventare sempre più forte. Se sono diventato un decente arrampicatore è soprattutto grazie a lui. A entrambi piace l'adrenalina e quindi ci divertiamo sempre un sacco assieme. Ovviamente ci sono un sacco di altre persone nella mia famiglia, così tante che è impossibile ricordarle tutte qui. Ad ogni modo, grazie a tutti voi.

Un’altra cosa da tener presente è che l’ambiente di lavoro è piacevole solo se hai dei buoni colleghi. A Innsbruck mi sono sentito come a casa fin da subito. Facevo parte di un gruppo fantastico che rendeva più facile la giornata di lavoro e meraviglioso il tempo libero. Lasciare Innsbruck è stato difficile perché è stato uno dei più bei periodi della mia vita, ma fortunatamente ho trovato dei colleghi fantastici anche a Trento, e grazie a loro il cambio è stato più facile. L’ufficio 141 è un posto felice e divertente in cui stare. Andare a lavoro ogni giorno è più facile se sai di condividere il tuo ufficio con delle persone così, anche se la tua sveglia suona già alle 6:15 di mattina.

Ultimi, ma non per questo meno importanti, ci sono i miei amici, che stanno al mio fianco fin dai tempi delle superiori. Abbiamo condiviso così tanti momenti insieme che è impossibile ricordarli tutti. Abbiamo un gruppo di amici favoloso, con tanti diversi interessi e passioni. Passare del tempo con loro è sempre divertente, e per qualunque problema loro sono sempre lì per te. Sono felice che il gruppo si stia ingrandendo e che un sacco di nuove persone entrino a fare parte della combriccola. Non ci si stufa mai con voi ragazzi. Grazie per essere un così bel gruppo di amici

Ci sono un sacco di persone che non ho elencato e che non potrei ringraziare personalmente senza aver bisogno di altre venti pagine. Tra di loro di sono colleghi di università, lo staff di università, i miei coinquilini di Innsbruck, le persone che ho conosciuto durante le conferenze, tutti i nuovi amici Gardenesi, e molti altri. Con loro ho condiviso dei momenti fantastici che non dimenticherò mai.

Grazie a voi tutti.