Cooperative Driving in Inter-Vehicular Communication Network

DIETI - Dipartimento Di Ingegneria Elettrica e Tecnologie dell'Informazione
Universitá degli Studi di Napoli Federico II
Italy.

A thesis presented for the degree of
Doctor of Philosophy in Computer Science and Control Systems Engineering

March 30, 2015

***Author*:**
Antonio Saverio Valente
***Tutor*:**
Prof. Stefania Santini
***Company Tutor*:**
PhD. Angelo Palladino

Contents

Abstract

Design, synthesis and validation of cooperative driving algorithm are the main actor of our treatment. Owing to the ever-increasing traffic demand, modern societies, with well-planned road management systems, and sufficient infrastructures for transportation, still face problems like traffic congestion and pollution. Intelligent transportation system supporting the driver during driving task are called ADA systems or ADAS. ADAS promise to increase the drivers safety and comfort with positive impact on traffic flow performance, emissions and fuel consumption. Examples of ADAS system are various forms of cruise control, lane keeping systems and collision warning systems. Recently the development of new communication protocols for vehicular environment based on WAVE/IEEE 802.11p standard has pushed industry and researchers toward the development of the concept of Cooperative Driving and Cooperative Adaptive Cruise Control systems. Cooperative driving control systems exploit the wireless communication as an additional sensor both to perceive the presence of neighbouring vehicles and to communicate their own presence and in-vehicle data. One of the most envisioned applications in cooperative driving systems is certainly platooning. Platooning concept can be defined as a collection of vehicles that travel together, actively coordinated in formation. Some expected advantages of platooning include increased fuel and traffic efficiency, safety and driver comfort. All vehicles within the platoon communicate with each other and exchange information in order to reach a common target. Here we aim to represent the platooning as a complex network, in which nodes represent the vehicles belonging to platoon and links model the existence/absence of communication among vehicles. In particular, we model platooning not only as a complex network, but as a delayed complex network. First we present how a group of interconnected vehicles can be modelled as a complex networks and then we treat the platooning problem first as an high-order consensus problem and then as synchronization problem.

The consensus goal is to regulate speed and relative position of each vehicle to that of the respective predecessor and of the leading vehicle. The idea is to analytically solve problem by designing a distributed control action depending on information received from the neighbouring vehicles

(within the transmission range). The control approach is able to counteract the presence of different time-varying delays introduced by the wireless vehicular communications, take into account the drivetrain dynamics and the heterogeneity of the platoon. First of all we give sufficient and necessary condition on the control gain that guarantee both exponential and global asymptotic stability. The stability of the proposed control strategy, has been shown by exploiting the Lyapunov-Krasovkii theorem for retarded functional differential equations, according a delay dependent approach. The stability conditions gives as additional result, an upper bound estimation of the maximum allowable communication delay that guarantee stability. Then we solve the platooning problem as a synchronization problem. Our target is to synchronize the dynamics of all agent of the platoon to the leader dynamics. The problem essentially consists in leader tracking manoeuvres. The synchronization goal is achieved here by using an appropriate adaptive distributed strategy, depending from local state variables as well as from the information received by the neighbouring vehicles. The control approach counteract the presence of different time-varying delays introduced by the wireless vehicular communications, and is robust to parameter variations. Also in this case we prove the stability of the proposed control strategy exploiting the Lyapunov-Krasovkii theorem.

Since in a real environment there is the unavoidable presence of time-delays, it is crucial, before the experimental validation of cooperative driving algorithms, of proper simulation tool taking into account of vehicular and control dynamics, not neglecting traffic patterns and the characteristics of the communication channel. These tools, known as VANET simulators, requires two types of simulation components: *i*) Network and *ii*) Mobility that in general are separate. Within a collaboration with the University of Trento, I have contributed to the development of PLEXE, a VANET simulator, that is the evolution of the well known Veins framework. In particular we have contributed to the formulation of a vehicle model to embed into PLEXE framework, that takes into account of transmission ratio and power of vehicles. All the proposed control strategy have been intensively validated and simulated with PLEXE simulator, both with the vehicle model and with default mobility model already present in PLEXE. The thesis is organized as follows.

In Chapter 1 we give first an overview on the Cooperative Driving systems. After an introduction about them, we focus on the main application considered during the dissertation: *i*) *Probe Vehicles*, *ii*) *Automated Fleets and Cooperative Driving*. Then we will give an overview on communication paradigms and enabling technologies in support of cooperative driving systems.

In Chapter 2 we focus on PLEXE, an open source and free to download VANET simulator allowing the simulation, validation and analysis of cooperative driving control strategy in mixed traffic scenario. After a brief description of PLEXE, we will illustrate the major enhancement and contributions given during this PhD work to develop a new vehicle dynamical model. We remark that these contributes have been embedded into PLEXE as part of a collaboration with the DISI - University of Trento.

In Chapter 3 first we present how a group of interconnected vehicles can be modelled as a complex networks and then we treat the platooning problem first as an high-order consensus problem and then as synchronization problem. The consensus goal is to regulate speed and relative position of each vehicle to that of the respective predecessor and of the leading vehicle. The idea is to analytically solve problem by designing a proper *local* action depending on the vehicle state variables and a *cooperative* action depending on information received from the neighbouring vehicles (e.g., within the transmission range). The control approach counteract the presence of different time-varying delays introduced by the wireless vehicular communications. We prove the stability of the proposed control strategy exploiting the Lyapunov-Krasovkii theorem and finally we will show the results obtained by exploiting PLEXE simulator. Then we treat the platooning problem as a synchronization problem. Our target is to synchronize the dynamics of all agent of the platoon to the leader dynamics. The problem essentially consists in leader tracking manoeuvres. The synchronization goal is achieved here by using an appropriate adaptive distributed strategy, depending from local state variables as well as from the information received by the neighbouring vehicles. The control approach counteract the presence of different time-varying delays introduced by the wireless vehicular communications. We prove the stability of the proposed control strategy exploiting the Lyapunov-Krasovkii theorem and finally we will show the results obtained using PLEXE simulator.

In Chapter 4 are resumed the company duties. We will present in particular the design and implementation of an On Board Unit, for collection and transmission of in-vehicle data to neighbourhood vehicle in an ITS scenario. To this purpose, a proper DSRC communication system has been designed and implemented. To validate the effectiveness of the designed On Board Unit a proper Hardware In the Loop platform has been developed to test the effectiveness of the control and communication strategy.

Finally, in Appendix A we present the maths instrument employed for our analysis: first we focus on the theorems used to show the stability of time delay systems; in second instance we present some theorems about the stability analysis of time-varying systems; finally we present some useful lemmas used in this thesis.

Chapter 1

Cooperative driving systems

Abstract

ITS that support the driver during driving task are called Advanced Driver Assistance (ADA) systems or more briefly ADAS. ADAS promise to increase the drivers safety and comfort with positive impact on traffic flow performance, emissions and fuel consumption. Development of ICT technologies allowed the design and implementation of different kind of ADAS application, exploiting sophisticate control strategies. An exemplar ADA system that has been introduced by the automotive industry is the CACC. CACC is one of the possible way to enable cooperative driving behaviour, exploiting wireless communication among vehicles. In what follow we will discuss about Cooperative Driving systems, enabling technologies and communication paradigm in vehicular application.

1.1 Introduction to Cooperative Driving

Owing to the ever-increasing traffic demand, modern societies with well-planned road management systems, and sufficient infrastructures for transportation, still face the problem of traffic congestion. This results in loss of travel time, and huge societal and economic costs and an increasing environmental impact [9]. To give new answers to the increasing mobility demand and other open issues, several solutions have been adopted. Some of these rely on the enhancement or on the construction of new infrastructure (like roads, highways, port, airport and so on), some others rely on the enhancement of the vehicular safety systems (like air-bags or safety belt). However, it is clear that building new infrastructure require several expensive actions [37] with the drawback of an increased environmental impact. For this reasons, solutions allowing a more efficient use of the existing infrastructure are aimed. Intelligent Transportation Systems (ITS) is a fairly new concept which gives a new vision of mobility that, thanks to the integration of Information and Communication Technologies (ICT) with traditional transport infrastructures, allows users to get more from transportation systems, in greater safety and with less environmental impact improving the overall efficiency. The idea of ITS, in road transports, rely on the possibility to connect among them vehicles with other vehicles or with central or de localized infrastructure, helping to find new solutions to problems like collision avoidance, fleet management, driver assistance etc. The goal thus became vehicles and infrastructure cooperate to perceive potentially dangerous situations in an extended space and time horizon [96].

Figure 1.1: ITS scenario: Vehicles communicates information each other [102]

From a general point of view, ITS that support the driver during driving tasks are called Advanced Driver Assistance Systems (ADAS) systems [118]. ADAS support drivers by strengthening their sensing ability, warning in case of error, and reducing the controlling efforts of drivers [73]. The building of automation systems to support rather than replace human drivers has become a trend in current intelligent vehicle research. To relieve the stress on drivers and reduce the risk of accidents, researchers have proposed several prototypes of cognitive cars that apply cognitive science to intelligent vehicle research [57, 127]. The cognitive car's aim is to achieve a more comfortable driving experience, monitor and detect the error/lapses of human drivers and correctly respond to avoid accidents. To reach this goal, different science field are involved, as automotive engineering, neuroscience, artificial intelligent and control theory and a strong interaction between this disciplines there exists.

The associated studies to ADAS were assumed to include the following [59]:

- perception enhancement;
- action suggestion;
- function delegation.

Clearly the research on the stimuli, decisions, and actions of human drivers is necessary to ADAS research area in order to understand what perceptions should be enhanced, what and when the action suggestion is required and how the function delegation should be appropriately implemented [73].

Driver perception enhancement can be considered as a mild ADAS, in which information about driving experience is directly provided to drivers who then can make decision/action themselves [73]. To enhance driver perception the concept of vehicles equipped with multiple sensor and vision systems, alternative to human eye, is one of major theme in this field because the most driving information is collected via eyes [73]. There are two vision enhancement: the inside-vehicle displays, which dynamically captures the scene in front of vehicle and the outside-vehicle lighting systems, which dynamically adjust the range of intensity of vehicle lights.

Action suggestion is considered as a moderate ADAS, that provides guidance to drivers but allows them to make decision [73]. There exist two types of action suggestion: *i*) visual display information and *ii*) voice navigation system.

Function delegation instead can be seen as an intensive ADAS, that can implements some detailed action with driver's command (e.g. steer-by-wire, brake-by-wire), or it can implements some action without driver's command (e.g. collision avoidance, lane departure avoidance), or also take all the necessary actions without driver's command (e.g. fully autonomous driving) [73].

Recently the development of new communication protocols for vehicular environment, pushed industry and researchers toward the development of the concept of Cooperative Driving. Cooperative driving systems exploit the wireless communication as an additional sensor both to perceive the presence of neighbouring vehicles and to communicate (in broadcast for example) their own presence and in vehicle data.

The idea of vehicles cooperating via wireless communication dates back to the 1980s [89], when Californias Partners for Advanced Transit and

Highways (PATH) program was established to study and develop vehicle-highway cooperation and communication systems [6, 107]. The basic idea is to enable the communication and the cooperation among neighbouring vehicles to safely reduce their mutual distance (thus leading to more vehicles without increasing the road capacity) and suppress traffic shock-waves (thus reducing pollutant emissions) [67]. The core of such cooperative driving systems is a set of algorithms deployed on the vehicles and controlling their motion based on the behaviour of the surrounding vehicles.

After twenty years of research and small-scale demonstrations, the diffusion of Global Navigation Satellite Systems (GNSS)s and wireless communication technologies for vehicular environment have stimulated research into cooperative driving applications, pushing automotive vendors to equip in the next future, these technologies into production vehicles [22]. This decade will see numerous deployments of infrastructure and in-vehicle technology to evolve beyond commercial applications and into safety and comfort domains.

Several benefit arise by exploiting high-accuracy positioning and wireless inter-vehicular communication technologies within active safety or ADAS systems. Current safety systems are reactive to sensor inputs and rely on real-time feedback with small time constants [21, 116]. With positioning and wireless communication technologies, cooperative driving applications introduce anticipatory behaviour to active safety systems enabled through autonomous sensors such as radars and cameras [21]. First, communication provides an unprecedented field of view of the driving environment and range for information sharing. Depending on the transmission frequency and power, wireless radio waves can travel long distances and through obstacles, this mean that line-of-sight is not required [21]. Second, the information that can be shared between vehicles, through Inter-Vehicular Communication (IVC), or with infrastructures, has greater quantity and higher quality than that measured or estimated remotely using autonomous sensors [21]. Data on each vehicles predicted route or past trajectory can be estimated by the vehicle, and then shared with others, rather than having each vehicle estimate this information for every other vehicle. Third, the cost of positioning and communication hardware is significantly less than autonomous sensors needed to cover the 360° envelope around the vehicle [21]. Finally, communications allow vehicles to coordinate manoeuvres for safety goals such as collision avoidance.

Communication and cooperation between vehicles offer the opportunity to develop many different applications [22]:

- *safety applications*: which aim to mitigate vehicular collisions such as rear ending;
- *mobility applications*: which look at increasing traffic flow through information sharing about road conditions;
- *comfort applications*: such as cooperative adaptive cruise control, which aim at reducing the drivers work load.

In this thesis we will focus mainly on two specific applications: *i*) *Probe Vehicles*, *ii*) *Automated Fleets and Cooperative Driving.*

Regarding *i*) vehicles have to be grouped or organized in fleets (a set of vehicles equipped with tracking devices for retrieving data [79]), while in *ii*) vehicles are organized as a platoon (a set of vehicles sharing information to reach a common objectives, e.g. a prefixed velocity [6,85]). In Chapter 4 we will describe an architecture designed to address the problem of creating and maintaining the composition of a fleet of vehicles. The solution proposed to address both problems is a system mainly composed of a vehicular On-Board-Unit (OBU), for the collection and dissemination of fundamental data from each vehicle in the fleet, and a centralized remote server for the analysis of data and management of fleets. In Chapter 3 we propose distributed control strategies for cooperative driving, with the aim to impose a common behaviour to a platoon of vehicles sharing information via wireless technologies. The presented cooperative driving control algorithm have been simulated and validated via software with PLEXE (a Vehicular Ad Hoc Network (VANET) simulator). A description of PLEXE and the major contributions given to the development of this framework are detailed in Chapter 2.

1.1.1 Probe Vehicles

The key idea is to give an active role to drivers, that hence contribute to collect information relevant for mobility with different aims (to monitor, for example, traffic congestion and pollutant emissions [7,121]). Until now, the collecting of such information has been mainly done on highways through traffic sensors, like cameras, loop detectors, ultrasonic sensors and air pollution sensors. More recently, the wide diffusion of modern tracking mobile devices, embedding the GPS technology, represents a new way to collect real-time information using transit vehicles as probes in urban context [113, 129]. A representative example of urban fleets of probes vehicles can be found in [62], where it is shown how a taxi company equipped their vehicles with GPS devices in order to collect and forward position, velocity and acceleration information to a remote server through the GSM network. In so doing, every vehicle is a part of an unique homogeneous fleet and acts as a probe that indirectly provides traffic level real-time measure. Following this vision, the thousands and thousands of data available on vehicle Controller Area Network (CAN) bus, produced by the on-board vehicles diagnostic systems, can be employed to get more accurate estimation of mobility parameters, like emissions or traffic. All this data can be also used for monitoring the vehicles state or for enabling some specific control application (like, as an example, lane changing assistance, ADAS, etc.).

Many research projects work in this direction, as an exemplar we mention the S2Move[1] Project, to whom I have personally contributed. The project aims to create an innovative platform that collect, analyses, process data from the field, and then return them in form of smart decisions to citizens. An S2Move user, (a driver or pedestrian) is not longer a mere consumer of a service but is the main actor of the platform providing new information that in turn are processed to be used by the whole community. The information are collected from many different devices as smart-phones and tablets, but also thanks to special probe vehicles that, opportunely equipped with on board hardware systems integrated directly into the cars, allows private vehicles and public utility to provide updated real time information on the state of urban mobility. The exchange of all the information is intended to produce a fundamental result: achieving an easy and sustainable mobility.

1.1.2 Automated Fleets and Cooperative Driving

Automated driving is an ADAS application that aims to help or to remove human drivers during some operation on vehicles in some specific driving conditions [112]. One of the most envisioned breakthrough of automated driving is represented by the Automated Highways System (AHS) that constitutes a key objective to increase the capacity of highways, reduce fuel consumptions

[1] S2Move: social innovation project funded by $MIUR(PON\ 04A3\_00058)$ - http://s2move.comics.unina.it:9000/

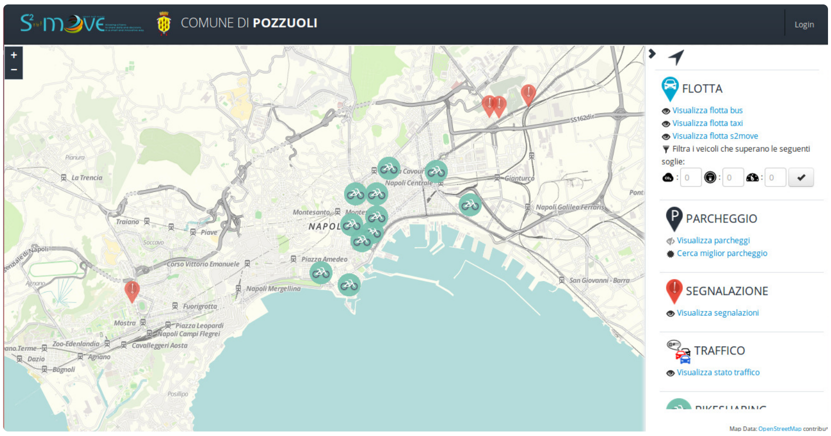


Figure 1.2: S2Move project: urban probes retrieving data for development of ITS application.

and increase drivers safety [110]. The concept of cooperative driving was first presented by the Association of Electronic Technology for Automobile Traffic and Driving (JSK) in Japan in the early 1990s. In AHS vehicles are organized in different automated platoons in presence of a driver [107], sharing common velocity and direction. Once again, the main idea is to get cooperation among vehicles, via wireless communication networks, to reach a common objective. The cooperative driving, which is an advanced form of the AHS, is defined as a set of automated vehicles, communicating each other, with a short inter-vehicle distance over a couple of lanes [114]. Taking moves from this consideration, the cooperative driving can be also defined as the coordinated motion of groups of vehicles "*platoon*", cooperating with each other to reach common target. Thanks to an appropriate inter-vehicle communication system, cooperative driving allows vehicles to perform safe manoeuvres as lane changing, merging or split (see Fig. 1.3). The communication between vehicles or between vehicles and road infrastructure enable vehicles and infrastructure to form a cooperative system where the users exchange information and cooperate to improve quality of travel experience. Forming platoon and operating them under automatic control (thus achieving an inter-vehicle separation smaller than the one guaranteed by human drivers, but safe) is a promising solution to increase road capacity, while decreasing, at the same time, traffic congestion [24, 51, 72, 122].

Another benefit, originated by cooperation, is that the aerodynamic drag is reduced (especially for heavy-duty vehicles) thereby increasing fuel economy

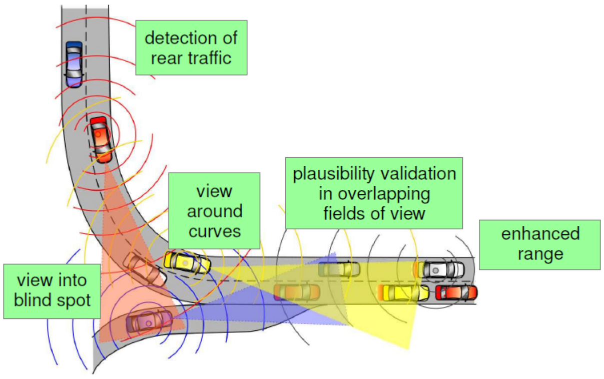

detection of rear traffic
plausibility validation in overlapping fields of view
view around curves
enhanced range
view into blind spot

Figure 1.3: Cooperation of vehicles in mixed traffic

and, consequently, reducing pollutants emissions [83]. As a consequence, the control of platoon of vehicles moving forward to a common destination has attracted, over the last two decade, a consistent interest both from scientists and practitioners. Usually control strategies solve a Cooperative Adaptive Cruise Control (CACC) problem, where the control objective is to impose a reference velocity to each vehicle in a platoon, (i.e. the velocity of the first vehicle in the platoon (leader vehicle)) with a "predecessor-following" architecture [56].

1.2 Communication Scenario and Technologies

Cooperative driving application requires efficient communications tools to forward data related to the vehicles state (position, velocity, acceleration, consumption, emission etc.) to other vehicle/pedestrian in the vehicle proximity, and/or to a remote systems. The variety of communication networks, such as 2-3-4G, WLANs IEEE 802.11a/b/g/p, and WiMAX, can be exploited to enable new services designed for passengers or safety applications, relying on the vehicular network itself [52].

Research project in Germany and Italy have evaluated GSM/GPRS and 3G network for IVC systems [8, 81]. The main motivation for using a mobile telephony standard for vehicular communication is that the infrastructure is already there and in the near future most vehicles in Europe will access to these networks. Other potential network access technologies are Bluetooth, specially for short-range communication.

In last decades, researchers has proposed in-vehicle communication technology based on a multi-access strategy, e.g. IEEE 802.11, UMTS/CDMA200, and Bluetooth [69]. Therefore they will probably require different network access technologies, addressing scheme, and protocols.

Starting from November 2004 a new promising protocol has been designed for vehicular networks: Wireless Access in Vehicular Environments (WAVE). WAVE represent one of the most promising communication technology based on IEEE 802.11p and IEEE P1609 standard (see Fig. 1.4). IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to add WAVE. In Europe, 802.11p was used as a basis for the ITS-G5 standard European Telecommunications Standards Institute (ETSI). IEEE 802.11p specifies the characteristics of physical and protocol layers for single-channel operations in way to support ITS applications. Therefore IEEE has also specified the IEEE 1609.x family of standards, which defines the functionality of the other WAVE protocol layers. The IEEE and the ETSI reserved respectively seven and five non-overlapping channels in the 5.85 GHz spectrum, each 10 MHz wide. One of these channels is the designated as Control Channel (CCH), while the remaining channels are Service Channel (SCH), as illustrated in [34]. The IEEE 1609.4 standard has been proposed to enhance the underlying 802.11p Medium Access Control (MAC) protocol with multichannel operations over a single-radio transceiver [120]. The multichannel operation are performed dividing the available access time between CCH and SCH into intervals of 50ms [120]. After each channel switch, a guard interval is used to separate CCH and SCH time interval. During the guard interval the channel is

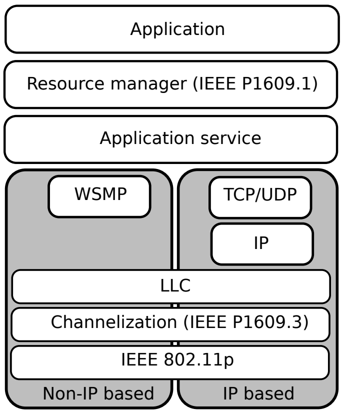


Figure 1.4: DSRC/WAVE standar: protocol stack.

sensed as busy, so that the transciever goes into back-off. The time division between CCH and SCH specified by IEEE 1609.4 could affect the IEEE 802.11p beaconing performance [34]. The CCH is used for the periodical dissemination of control information. The procedure used by the CCH for the dissemination of this control information is denoted as beaconing and the message are denoted as Cooperative Awareness Message (CAM) (ETSI's - European standard) or Basic Safety Message (BSM) (Society of Automotive Engineers (SAE)'s - American standard). Moreover, the CCH is also used for the dissemination of services advertise: traffic and safety related information event messages. The SCHs instead are used to disseminate non-critical information for infotainment applications [34, 120]. WAVE support both IP and non-IP based applications. The non-IP based applications, for example, emergency warning messages, are supported by the WAVE short message protocol (WSMP). IEEE 802.11p is the support for applications priorities, which means high priority data, for example, related to collision avoidance, will be transmitted with minimum latency [69].

Recently, worldwide ISO TC204/WG16, has produced a series of draft standards known as Continuous Air-Interface, Long and Medium Range (CALM). The goal of CALM is to develop a standardized networking terminal that is capable of connecting vehicles and roadside systems continuously and seamlessly [69, 88]. The CALM architecture is based on an IPv6 convergence layer that decouples applications from the communication infrastructure. A standardized set of air interface protocols is provided for the best use of resources available for short, medium and long-range, safety critical communications, using one or more of several media, with multipoint (mesh) transfer.

To allow communication among nearby vehicles or between vehicles and nearby fixed roadside equipment, different architectural solution for creating vehicular networks were proposed. Such architectures have to guarantee two main communication paradigm:

- a pure wireless Vehicle to Vehicle (V2V), allowing vehicular communication with no infrastructure support;
- an hybrid Vehicle to Infrastructure (V2I) architecture that does not rely on fixed infrastructure in a constant manner, but can exploit it for improved performance and service access when it is available.

Actually the V2I architecture implicitly includes V2V communication. The Car-to-Car Communication Consortium (C2C-CC), a non-profit and an industry driven organisation initiated by European vehicle manufacturers and supported by equipment suppliers research organisations and other partners, proposed a reference architecture for vehicular communication network. In this architecture they distinguish between three domains: *i*) in-vehicle, *ii*) ad-hoc and *iii*) infrastructure domain [111].

The in-vehicle domain refers to a local network inside of which each vehicle is logically composed of an OBU and Application Unit (AU)s. An OBU is a device in the vehicle having communication capabilities (wireless and/or wired), while an AU is a device that execute a set of application, using OBU's communication capabilities. An AU can be an integrated part of the vehicle or a portable device such as a laptop or Portable Device Assistant (PDA) that can dynamically attach to OBU or detach from OBU itself. The two unit are usually connected with a wired connection; also wireless connection through Bluetooth, WUSB or UWB is possible. This distinction between AU and OBU is logical, and they can also reside in a single physical unit, in Sec. 4.1 we will give a more detailed description about technological features of an OBU.

The ad-hoc domain, or VANET, is composed of vehicles equipped with OBUs and stationary units along the road, called Road-Side Unit (RSU)s. The OBU of every vehicle form a Mobile Ad Hoc Network (MANET), where an OBU is at least equipped with a short range wireless communication device dedicated for road safety, and potentially with other optional communication devices. The primary role of an RSU is the improvement of road safety, by executing special applications and by sending, receiving or forwarding data in the ad hoc domain in order to extend the coverage of the ad-hoc network. OBUs and RSUs can be seen as nodes of an ad-hoc network, respectively mobile and static nodes. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. As a result, RSUs may allow OBUs to access the infrastructure. The ad-hoc domain is often referred as VANET, i.e.: *a novel class of wireless networks that have emerged thanks to advances in wireless technologies and the automotive industry. Vehicular networks are spontaneously formed between moving vehicles equipped with wireless interfaces that could be of homogeneous or heterogeneous technologies. These networks, also known as VANETs, are* *considered as one of the ad-hoc network real-life applications, enabling communications among nearby vehicles as well as between vehicles and nearby fixed equipment, usually described as roadside equipment* [88].

Over RSU an another infrastructure domain access exist: the Hot Spot (HP). OBU's can also communicate with internet through public, commercial, or private hot spots. In the absence of RSUs and hot spot, OBU's can use radio networks (GSM, GPRS, UMTS, WiMax and 4G) if they are integrated in the OBU itself [87].

These types of communication paradigm allow a number of deployment options for vehicular networks beyond cooperative driving systems.

1.3 Characteristics of Cooperative Driving system

A cooperative driving system has little, no or mostly unpredictable control over the behaviour of other vehicles in the vicinity. Expected behaviour may be indicated by regulations and legislation, but from a purely technical point of view, it is a safety hazard to assume that all other systems will conform to and abide by the rules: the only certainty is the ability to control the ego vehicle and broadcast information about it [10].

Despite this, it is the responsibility of all cooperative driving systems to maintain the integrity of the traffic flow in the vicinity. Functional characteristics of a cooperative driving system can be roughed out as follows [10]:

- a cooperative driving system is characterized by distributed, hierarchical control. The high level control structure takes decisions and compute set point for lower level controller;
- in cooperative driving systems the perception of the surrounding environment is limited and entrusted to the input sensors. The characteristics of designed applications are strictly tied to the sensors used to obtain informations;
- a cooperative driving system is closely related in concept to an autonomous system. This implies that there must exist a system component that know at any time what function are available and who is performing them;
- a cooperative driving system needs a Human Machine Interface (HMI) that can be used to make the human driver aware of what the system

is doing.

1.3.1 Architecture for Cooperative Driving Applications

Automated vehicles, under the cooperative driving, drive like the migration of birds or a group of dolphins; the formation of birds in the migration is aerodynamically efficient, and dolphins swim without collision while communicating with each other. The cooperative driving, simulating the formation of birds or dolphins, will contribute to the increase in the road capacity as well as in the road traffic safety [115]. To get such vision of cooperative driving, a suitable technological architecture have to designed. A possible solution that provide a clear view of the cooperative driving to make the design transparent and efficient can be found in [115]. The proposed architecture, consists of three layers as shown in Fig. 1.5. The vehicle control layer and the vehicle management layer are on each vehicle, and the traffic management layer, which is common to all the vehicles and shared by them is on the infrastructure.

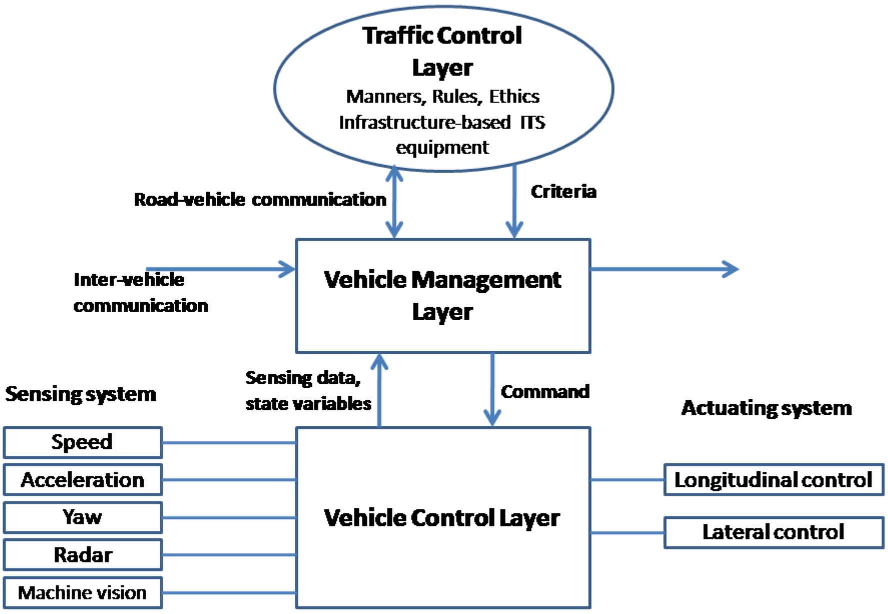


Figure 1.5: The architecture for the cooperative driving of automated vehicles

The functions of the vehicle control layer are to sense the conditions and states ahead of the vehicle and to activate the lateral and longitudinal actuators. The layer outputs sensing data and vehicle state variables to the vehicle management layer and receives commands of the steering and vehicle speed from it. Each vehicle may have its individual vehicle control layer.

The vehicle management layer determines the movement of each vehicle under the cooperative driving with the data from the vehicle control layer, those received from neighbouring vehicles through the inter-vehicle communications, and from the traffic control layer through the road-vehicle communications. The criteria of the movement comes from the traffic control layer.

The traffic control layer has two parts: physical part, that includes the infrastructure-based ITS equipment like sign boards, traffic signals, and the road-vehicle communications, and a logical part that includes common sense, laws, rules, manners, and ethics in the human society. Within the two parts, a criteria that must be common to neighbouring vehicles will be found and sent to the vehicle management layer in each vehicle.

1.3.2 Control Strategies for Cooperative Driving

Perhaps the control strategy precursors of cooperative driving system, introduced by the automotive industry, is the Adaptive Cruise Control (ACC) system. ACC is considered to be the successor of the conventional cruise control Cruise Control (CC). A vehicle with CC is able to maintain a preselected speed if no vehicle is upfront. ACC is a radar-based system which is designed to enhance driving comfort and convenience by relieving the driver of the need to continually adjust his or her speed to match that of a preceding vehicle. The system slows down the speed when it approaches a vehicle with a lower speed and the system increases the speed to the level of speed previous set when the vehicle upfront accelerates or disappears (e.g. by changing lanes) [124]. Recently, V2V communication have pushed the ACC system into a more sophisticate system, called CACC. Each vehicle within the cooperative driving system is equipped with on-board sensors measuring position, velocity, acceleration. Such set of measurements requires Inertial Measurements Units (IMU), Global Positioning Systems (GPS) and radars, which are commonly available on road vehicles. Each vehicle is also equipped with wireless V2V communication hardware to share information with its neighbours and receive reference signals. Thanks to the information of neighbours vehicles CACC controller will be able to anticipate problems better, enabling it to be safer, smoother and more reliable in response. Wireless communication is used to regulate speed and distance between vehicles with CACC. In CACC, automated information on variables such as speed and acceleration is obtained from other vehicles on the road. The corresponding controller ensures that any changes in speed by the driver in front of you are immediately registered in the cooperative vehicle. These control system are usually deployed on vehicle management layer and traffic control layer of the architecture in Fig. 1.5.

However, most of the CACC controller presented in literature does not cope communication failure/impairments and network delay. Furthermore the cooperative strategy rely on a fixed communication paradigm among vehicles that, in many cases, take care only of the front vehicle informations [17,125]. As an example, the CACC designed in [98] considers data from the front vehicle only, while the one in [100] exploits data from the leader as well. What current approaches assume is a static control topology, which means that the design of the controller is based on a fixed communication pattern. When such communication pattern changes due to, for example, network impairments, the CACC is not able to safely control the platoon any more.

To overcome this problem flexible control system, reconfigurable on the basis of the actual communication capabilities, have to be designed. It has been shown that, the design of coupling protocols for the coordination of a group of agents exchanging information in the presence of limited and uncertain heterogeneous communication delay can be seen as a consensus problem or a synchronization problem [32,95]. The proposed approach is specifically designed to take into account communication logical topology, as well as impairments as delay and losses. In this sight, cooperative driving can be represented as a dynamical network where:

1. nodes are the vehicles characterized by their own dynamics;
2. edges model the presence/absence of communication between agent;
3. the structure of inter-vehicle communication is encoded in the network topology.

The control input acts on every vehicle of the cooperative driving systems and it depends on the state variables of vehicle itself and on information received from neighbouring vehicles through the communication network [32,61,67,95,114].

Part of this research work (see Chapter 3), has been focused on the analysis and solution of the problem of creating, maintaining and coordination of a platoon of vehicles, treating it as the problem of achieving consensus or synchronization in a network of dynamical agents in the presence of time-varying communication delay [32, 92, 94].

The communication time-delay is explicitly modelled in order to give a more realistic representation of the cooperative driving systems. So we can see cooperative driving systems, not only as complex network in which the vehicles represent the nodes of network and links the presence of communication between these nodes, but as complex delayed network in which the communication between agent is affected by a time delay. From a control viewpoint, the main goal for vehicles is to form a platoon and then to regulate speed and relative position of each vehicle to that of the respective predecessor and of the leading vehicle, according to a desired spacing policy that could be constant or may vary according to a *time headway* [100].

1.4 Vehicular Ad Hoc Network Simulators

In automated driving application, it is clear that the efficiency of V2V wireless communication has a great impact on the effectiveness of the local control action that needs, to work properly, information coming from other vehicles in a reliable and deterministic way. CACC algorithms needs frequent and up-to-date information (around 10 Hz [98]) about vehicles in the platoon in order to avoid instabilities, which might lead to collisions. For this reason, it is crucial to understand and analyse if and when the communications requirements can be satisfied by the network protocols. More in depth, there is the necessity to look if the current network protocols (e.g. Dedicated Short Range Communications / Wireless Access in the Vehicular Environment (DSRC/WAVE) 802.11p based) can supports V2V based control strategies, standard or innovative, and how some fundamental parameters like beaconing frequencies, delay, packet loss, adaptive power transmissions strategy, channel access methods, affect the V2V based control algorithm performance.

In this vision, to validate new vehicular applications (or more simply to verify standard CACC algorithm performance), an appropriate VANET simulation framework must be employed [93]. Such framework must allow to simulate communication in real roadway scenario giving the opportunities to model real traffic [93], objects and buildings, to take into account all fading phenomena associated to the radio communication. Note that, deploying and testing VANETs on real vehicles involves high cost and intensive work, and often involve large and heterogeneous scenarios. Due to the high expenses and dangers of testing new technologies for transportation in real world, simulation plays an important role to find out the beneficial and effective technologies before implementation [93].

This implies that, to simulate the performance of new ADAS application based on VANET communication environment, proper tools, taking into account vehicular dynamics and control, traffic patterns and the characteristics of the communication channel and protocols are required. For this reasons this tools requires two types of simulation components: *i*) Network and *ii*) Mobility; that in most cases results separates [93]. This need is even greater if we consider that the features of these application are often safety critical and a full and deep simulation and validation phase is a fundamental milestone. Another issue, that require a flexible and realistic simulation environment, is given by the lack of a precise common standards for wireless application in a vehicular environment. Many national and international projects in government, industry, and academia are working in this direction.

Different kind of simulator are available to reproduce the behaviour of a VANET; in particular [93] give a classification into three main classes: *a*) Network generators, *b*) Traffic simulators, *c*) Software that integrates the potentiality of *a*) and *b*) which can simulate both mobility and network. In the latter type, traffic simulator generates the vehicular mobility traces to be used in the network simulator as input. The network simulator in turn calculates all the parameters of the wireless network, send and receive packets, elaborate traffic data information, model the channel and the network protocol behaviour.

There are many different VANET simulators available e.g. TraNS, MOVE, NCTUns etc. that are mainly based on the well-known network simulator NS3 or NS2. The VANET simulator that we have considered in this work, is Veins ant its evolution PLEXE, to which is dedicated the following chapter.

Chapter 2

Plexe: Platooning extension for Veins

Abstract

Veins is a VANET simulator, open source and free to download, that allow users to embed new algorithm, test and validate ITS applications. Recently an extension of Veins has been released: PLEXE*. It is an effective VANET simulator allowing simulation, validation and analysis of cooperative driving control strategy in mixed traffic scenario. After a description of Veins, we will discuss about its extension for platooning (*PLEXE*), illustrating the major enhancement and contributions given during my PhD work. This contributes have been embedded into* PLEXE *as part of a collaboration with the DISI - University of Trento.*

2.1 Veins

Veins is a VANET simulation framework that provides coupled network and road traffic simulation using well-established simulators from both communities (transportation and networking). Veins, realized by Christoph Sommer *et al.* [108], is an open source IVC simulation framework composed of an event-based network simulator (*OMNeT++*) and a road traffic micro simulator (*SUMO*) by extending each with a dedicated communication module (*TraCI*).

The overall structure of Veins is depicted in Fig. 2.1, where is possible to distinguish its software architecture.

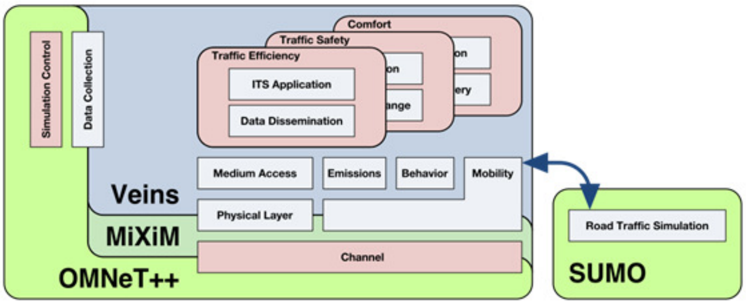


Figure 2.1: Veins functional block [84]

Veins extends the OMNeT++ network simulator by providing a complete vehicular communication stack, based on IEEE 802.11p and tailored channel models, together with a way of modelling realistic node mobility based on the road traffic simulator SUMO [103]. For this reason it couples both the network and the mobility simulator by creating a network node in OMNeT++ for each vehicle travelling in SUMO. Each OMNeT++ node can be associated with a network stack which includes an IEEE 802.11p wireless network interface, plus a beaconing protocol and one or more applications running on top of it. Each time a vehicle moves, Veins replicates the movement in the corresponding OMNeT++ node by updating the mobility model. To this aim both simulators are running in parallel, connected via a TCP socket, exchanging commands, as well as mobility traces. OMNeT++ is an event-based simulator, so it handles mobility by scheduling node movements at regular intervals. This fits well with the approach of SUMO, which also advances simulation time in discrete steps.

As can be seen in Fig. 2.2, the control modules integrated with OMNeT++ and SUMO are able to buffer any commands arriving in-between time-steps to guarantee synchronous execution at defined intervals. At each time-step, OMNeT++ would then send all buffered commands to SUMO and trigger the corresponding time-step of the road traffic simulation. Upon completion of the road traffic simulation time-step, SUMO would send a series of commands and the position of all instantiated vehicles back to the OMNeT++ module. This allows OMNeT++ to react to the received mobility trace by introducing new nodes, by deleting nodes that had reached their destination, and by moving nodes according to their road traffic simulation counterpart [108]. After processing all received commands and moving all nodes according to the

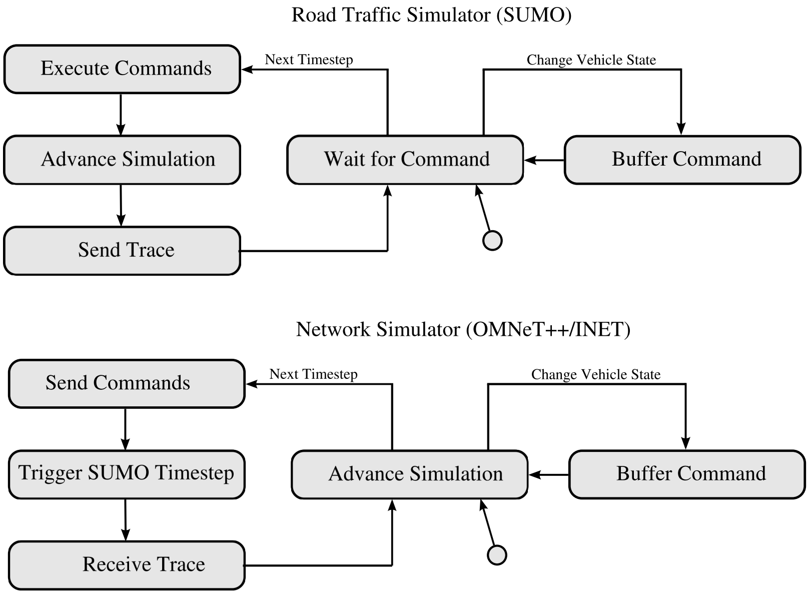


Figure 2.2: State machines of road traffic and network simulator communication modules [108].

mobility information, OMNeT++ would then advance the simulation until the next scheduled time-step, allowing nodes to react to altered environmental conditions, i.e., the IVC is influencing their speed and routes [108].

The protocol for coupling both simulators has been standardized as the Traffic Control Interface (TraCI). In other words Veins uses the so-called TraCI-API of SUMO to communicate via a TCP connection with OMNeT++ and vice verse. Actually Veins has already implemented a basic functionality of the TraCI commands, that is possible extend furthermore. The TraCI interface allows users to bidirectionally-couple the simulation of road traffic and network traffic in very easy way. Movement of vehicles in the road traffic simulator SUMO is reflected in movement of nodes in an OMNeT++ simulation. Nodes can then interact with the running road traffic simulation, e.g. to simulate the influence of IVC on road traffic. Using a simple request/response protocol, road traffic in SUMO can be influenced by OMNeT++ in a whole number of ways. Most importantly, time-steps are generated to advance the simulation in SUMO [108]. Illustrated in Fig. 2.3 are the alternating phases of coupled simulation which result from this approach.

In the first phase, commands are sent to SUMO, and in the second phase,

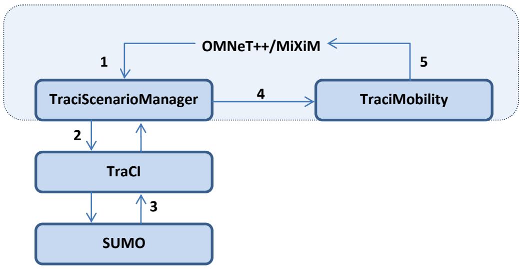


Figure 2.3: OMNeT++ and SUMO coupling schematic in Veins.

their execution is triggered and the resulting mobility trace received. This way, both simulators are tightly coupled and SUMO is only able to perform a simulation step after all events within a time step have been processed in the network simulation. The network simulator advances the road traffic microsimulation only at fixed intervals. That means that the granularity of these intervals needs to be sufficiently fine-grained to obtain realistic results. From a performance point of view, this is not a problem because the road traffic microsimulation can be processed much faster compared to the simulation of wireless networks [108]. In the follow we will describe briefly SUMO, OMNeT++ and MiXiM as integrating part of Veins.

2.1.1 SUMO: traffic simulator

The necessity of accurate traffic simulator has grown in the last years, since they pose as useful tools for evaluate ITS application performance. There are several approach to study traffic flow and as consequence there are different traffic models [18]:

1. dynamic macroscopic traffic flow models;
2. dynamic microscopic traffic flow models;
3. mesoscopic models.

Dynamic, time-variant, macroscopic models describe the evolution of traffic over time and space using a set of differential equations involving variables such as flow, density, speed and speed variance. In microscopic models, traffic is described at the level of individual vehicles and their interaction with each other and the road infrastructure. Normally this behaviour is captured in some set of rules which determine when a vehicle accelerates, decelerates, changes lane, but also how and when vehicles choose and change their routes to their destinations. Mesoscopic models fill the gap between the aggregate level approach of macroscopic models and the individual interactions of the microscopic ones [18].

In the following we will focus briefly on SUMO (*Simulator of Urban MObility*) the microscopic traffic simulator embedded into Veins. SUMO is an open source traffic simulation developed by the German Aerospace Center (DLR) back in 2001. SUMO offer to users different features giving the possibility to: *i*) create or import a road network topology from different source formats, *ii*) employ the routing utilities from various input sources, *iii*) to employ a *remote control* interface (TraCI) to adapt the simulation online. Each vehicle is given explicitly, defined at least by an identifier (name), the departure time, and the vehicles route through the network. If wanted, each vehicle can be described more detailed and with own designed model. For each vehicle is possible to define the departure and arrival properties, such as the lane to use, the velocity, or the position; furthermore each vehicle can get a type assigned which describes the vehicles physical properties and the variables of the used movement model. Additional variables allow the definition of the vehicles appearance within the simulations graphical user interface and the pollutant or noise emission classes [78]. To simulate the road traffic SUMO requires the representation of road networks and traffic demand. For this purpose SUMO offer the possibility to generate road networks both using an application named *netgen* or importing a digital road map. The road network importer *netconvert* allows also to read networks as shape files or as Open Street Map file.

Probably the most popular application for the SUMO suite is modelling traffic within research on C2C-CC communication. To this purpose, SUMO is often coupled to an external network simulator, such as ns2, ns3 or OMNeT++ via TraCI interface.

2.1.2 OMNET++: network simulator

Network simulators are used by people from different areas such as academic researchers, industrial developers, and Quality Assurance (QA) to design, simulate, verify, and analyse the performance of different networks protocols. Some examples of famous network simulation software are NS (open source), OPNET (proprietary software), NetSim (proprietary software) and OMNeT++ [14].

OMNeT++ is an object-oriented modular discrete event simulator. The name itself stands for Objective Modular Network Testbed in C++. OMNeT++ is mostly applied to the domain of network simulation, given the fact that with its INET package it provides a comprehensive collection of Internet protocol models. The most common use of OMNeT++ range from simulation of computer networks to Multiprocessors and distributed hardware systems and so on. OMNeT++ was developed at the Technical University of Budapest, Department of Telecommunications and is licensed under the own Academic Public License, which allows the GNU Public License like freedom but only in non-commercial settings [48].

OMNeT++ provides the infrastructure and tools for writing simulations but it isn't a ready to use Network Simulator by itself. A simulation model in OMNeT++ consists of a well designed arrangement of reusable *modules* which are connected and communicates each other via *gates*. The smallest building block in a simulation model is called the *simple module*. Simple modules are written in C++, and corresponds to a class definition. Simple modules, can be made to behave as desired and can be assembled into larger components (*compound modules*) and models using a high-level language (NED Language-*NEtwork Description*). Modules communicate with *messages* which in addition to usual attributes such as time-stamp may contain arbitrary defined data. Simple modules typically send messages via *gates*, but it is also possible to send them directly to their destination modules. Gates are the input and output interfaces of each modules: messages are sent out through output gates and arrive through input gates. An input and an output gate can be linked with a *connection*. Connections are created within a single level of module hierarchy: within a compound module, corresponding gates of two sub-modules, or a gate of one sub-module and a gate of the compound module can be connected. Each module can have *parameters*, that are mainly used to pass configuration data to simple modules, and to help define model topology. The values of these parameters (strings, numbers, truth table, etc) can be assigned and modified in the corresponding *NED* file, or configured in a file called *omnetpp.ini*. Since this file has its own syntax, you can change the values of parameters without having to recompile the model. All these features are provided as a networking simulation tool based on Eclipse that allow developers to design and define application and modules in a simple way.

MiXiM (Mixed Simulator)

Although OMNeT++ provides a powerful and clear simulation framework, it lacks of direct support and a concise modelling chain for wireless communication. MiXiM (*Mixed Simulator*) is an OMNeT++ modelling framework that provides detailed models of various layers of the OSI model along with accurate models for radio wave propagation and interference estimation, radio transceiver power consumption and wireless MAC protocols [30]. It also has a vast library of networking protocols that can be incorporated into any simulation scenario. Since it follows a modular hierarchical software structure, where simple modules can be composed to form more complex compound modules, a great deal of flexibility is available when it comes to developing new compound modules. Also, user defined simple modules can be added to the framework, allowing to model custom compound modules. Veins simulation framework comes as part of MiXiM, adding support for IEEE 802.11p, IEEE 1609.4 and IEEE 1609.3 (WSMP), allowing simulation of networks with up to 1000 nodes (cars).

This features has been necessary for the implementation of this dissertation since the application protocol is implemented as a simple module and then incorporated into a more complex compound module defining the vehicle, (representing a the node in the VANET), in the simulation scenario.

TraCI

TraCI stands for Traffic Control Interface, and this is what gives the outside world access to an already running simulation in SUMO in real time. It uses a TCP based client/server based architecture to provide access to SUMO. As part of the Veins project, the TraCI Scenario Manager module in OMNeT++ was introduced as part of the MiXiM framework. This gives us the ability to couple OMNET++ and SUMO together as shown in Fig. 2.3. It enables a network simulation running in OMNeT++ to send TraCI commands which control the traffic simulation in SUMO. The TraCI protocol explains in detail the messaging and message format to be used. By using this interface, Veins queries SUMO about current “traffic” status (e.g., number of vehicles, their position and speed, etc.), and it is able to modify the traffic dynamics, for instance by changing the route a vehicle SUMO is travelling on, or its acceleration.

2.2 Plexe: Platooning Extension for Veins

Plexe is an extension of Veins simulator framework, introducing several new features that enable the realistic simulation of cooperative driving and platooning applications. Based on Veins, Plexe allows to embeds into their structure all the algorithm and the manoeuvres to form control and maintain a platoon of vehicles. The main characteristics of Plexe can be listed as follow:

- is a free and open source tool;
- allow users to realize simulation of cooperative driving scenario thanks to the realistic model of wireless and road network;
- is kept constantly updated and improved, thanks to the feedback of the community;
- extensibility to different road traffic scenario or different new control model;
- realism of simulation in mixed traffic scenario where human-driven vehicles coexists with controlled one.

2.2.1 Plexe structure

Plexe further extends the interaction through the TraCI interface in order to fetch vehicles' data from SUMO to be sent to other cars, and to be used by the platooning protocols and applications. Data received by vehicles in Plexe can be fed to the cooperative driving algorithm implemented in SUMO. Platooning beaconing protocols as well as the application logics are realized in the OMNeT++ framework, while the actuation of the applications decisions together with part of the application logics are implemented in SUMO [103]. Fig. 2.4 depicts the schematic overview of the extended simulation framework.

The efforts to implement a correct platooning model unfurls in two directions: *i*) the implementation of platooning capabilities and elementary manoeuvres for vehicles, which mainly requires changes and extensions in SUMO; *ii*) the implementation of protocols to support the applications and the application logic itself in OMNeT++/Veins plus minor changes to enhance the bidirectional coupling [103]. The version of Plexe we used in this thesis work is based on Veins 2.2 and SUMO 0.17.1. This version

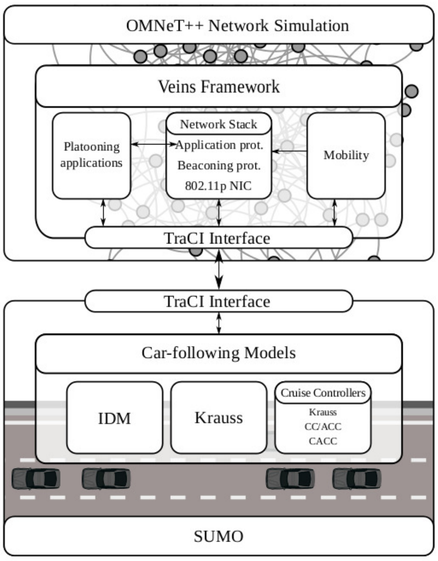
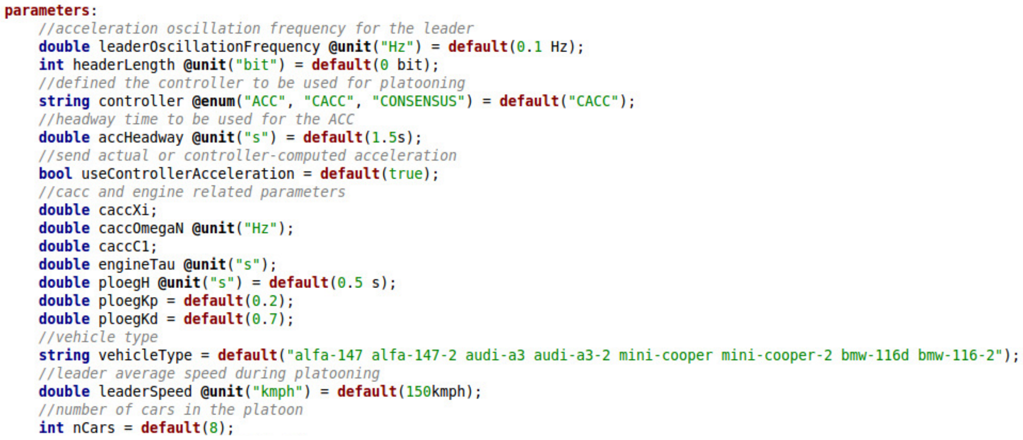


Figure 2.4: Plexe functional structure [103].

available on the web site[1] has already embedded different car-following model which enables both longitudinal control based on open or closed-loop control of the acceleration, and a simplified transversal control (i.e., steering) to appropriately change lanes and obey platoon dynamics. In particular, this new car-following model in SUMO makes the longitudinal controllers generically called CC available and accessible via TraCI. SUMO car-following models are conceived to mimic the behaviour of drivers e.g. Intelligent Driver Model (IDM), or the Krauss car-following model. This car-following models are particularly useful to let the car join and leave the simulation. For example, the car might enter the highway using an on-ramp driven by a human [103]. Then, the automated controllers can be switched on to drive the car until the desired exit, and can be turned off (giving the control back to the driver) to exit the highway using an off-ramp. The CC model present in the version of Plexe includes standard CC/ACC and one advanced CACC, plus the engine actuation lag, while another advanced CACC is under testing and will be available soon [32]. Through the TraCI interface it is possible to access the model, changing its behaviour and retrieving different information. Parameters, desired speed and coefficients for each control algorithm are editable through the OMNeT++ configuration *.ini* file or by default in *.NED*

[1] Plexe website: `http://plexe.car2x.org/`

interface (see for an exemplar case Fig. 2.5).



```
parameters:
    //acceleration oscillation frequency for the leader
    double leaderOscillationFrequency @unit("Hz") = default(0.1 Hz);
    int headerLength @unit("bit") = default(0 bit);
    //defined the controller to be used for platooning
    string controller @enum("ACC", "CACC", "CONSENSUS") = default("CACC");
    //headway time to be used for the ACC
    double accHeadway @unit("s") = default(1.5s);
    //send actual or controller-computed acceleration
    bool useControllerAcceleration = default(true);
    //cacc and engine related parameters
    double caccXi;
    double caccOmegaN @unit("Hz");
    double caccC1;
    double engineTau @unit("s");
    double ploegH @unit("s") = default(0.5 s);
    double ploegKp = default(0.2);
    double ploegKd = default(0.7);
    //vehicle type
    string vehicleType = default("alfa-147 alfa-147-2 audi-a3 audi-a3-2 mini-cooper mini-cooper-2 bmw-116d bmw-116-2");
    //leader average speed during platooning
    double leaderSpeed @unit("kmph") = default(150kmph);
    //number of cars in the platoon
    int nCars = default(8);
```

Figure 2.5: Example of a NED configuration file.

Obviously, when a CACC system is on, it is possible to feed it with data exchanged via IVC. Moreover, the interface includes a method to feed the controller with data about any vehicle in the platoon, because there exist controllers that can use data received from arbitrary vehicles to improve their performance as demonstrated in [32]. User can choose which of the different implemented control strategy or car-following model is driving the car, i.e., the human behavioural model, any of the CC, ACC, or CACC, or any other controlling model implemented in the simulator customized by the user. For testing and safety verification purposes, it is possible to set different behaviours for a platoon leader, e.g., a constant acceleration. Furthermore Plexe provide a basic network (extensible) stack where each car is provided with an IEEE 802.11p network interface card, a basic protocol for message dissemination, and an application layer running directly on top of the message distribution. The idea of the protocol layer is to implement the communication strategy to share the information among the vehicles in the platoon. The construction of beaconing strategy and applications (e.g. special leader manoeuvres like tip-in/tip-out or follower manoeuvres like joint/leave platoon) relies on already implemented C++ base class named *BaseProtocol* and *BaseApp* taking care of loading simulation parameters, or passing data to the CACC via TraCI, logging of statistics. In this way inherited classes can focus just on the implementation of the desired behaviour. For what concerns human-driven vehicles, it is possible to set up standard SUMO traffic flows, which will be part of the simulation as well. Human-driven vehicles can be used to create road traffic (i.e., mobility) disturbances, or network interference by running different applications that compete for the channel with platooning cars [103].

2.3 Ideal Channel model in Plexe: bursty packet loss

To investigate the robustness and the effectiveness of the designed control algorithm (presented in Chapter 3) in the presence of communication failure, we have programmed PLEXE to reproduce two different ideal channel models: *Gilbert-Elliott* and *Bernoulli* model.

2.3.1 Bernoulli loss model

A very simple model for packet drop phenomenons is the Bernoulli loss model. It is characterized by a single parameter represented by the loss rate r that correspond to the probability to loss a packet. The model can be simply implemented by picking a random number for each packet, and deciding whether it is lost comparing the loss rate r with the value of the picked number. It is important to say that in the Bernoulli model the packet loss probabilities are independent of each other, therefore the probability of j packet losses within a series of n consecutive packets, $P(j,n)$, is then given by the binomial distribution [133]:

$$P(j,n) = \binom{n}{j} r^j (1-r)^{n-j},\ n \geq 1,\ 0 \leq j \leq n, \tag{2.1}$$

where

$$\binom{n}{j} = \frac{n!}{j!(n-j)!} \tag{2.2}$$

is the binomial distribution. Such a model does not characterize burst packet loss adequately, for this reason we have developed also a Gilbert Elliot channel model.

2.3.2 Gilbert Elliot in Plexe

The Gilbert Elliot model has been implemented to simulate errors and bursty packet loss in the transmission channels [35]. We consider the 2-state Markov approach as introduced by Gilbert and Elliott, which is widely used for describing error patterns in transmission channels [53]. The Gilbert-Elliot model is composed by a Markov chain with two states called *GOOD* (G) and *BAD* (B). In state GOOD the noise/errors occurs with a low probability, while in state BAD the presence of burst noise into the transmission assumes considerable level [43].

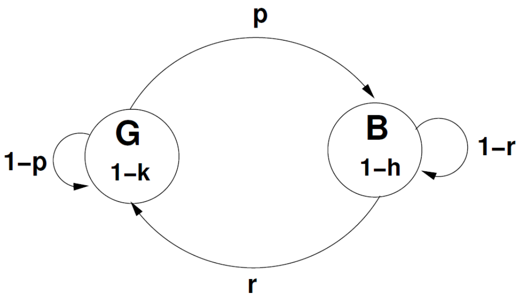


Figure 2.6: Transition diagram for the Gilbert-Elliot model.

To characterize the data loss processes the automaton shown in Fig. 2.6, is used. In particular we interpret an event as the arrival of a packet and an error as a packet loss. The parameter p and r represent the transition probabilities, and are so defined [53]:

$$\begin{aligned} p &= P(q_t = B|q_{t-1} = G), \\ r &= P(q_t = G|q_{t-1} = B). \end{aligned} \tag{2.3}$$

Where q_t is the state at the time t. In our case p and r are modelled with the desired probability distribution, for example the exponential probability distribution, whose p.d.f. function is given by:

$$f(x) = \begin{cases} -\lambda \exp^{-\lambda x} & x \geq 0 \\ 0 & x \leq 0 \end{cases} \tag{2.4}$$

In Eq. (2.4), λ represents the failure rate. Algorithm 1 report the pseudo-code describing 2-state Markov chain model described formerly.

Algorithm 1: Pseudo-code of Gilbert Elliot Markov chain model

```
Data: stateChange,GOOD,BAD
Result: Gilbert Elliot state change
if (actual state is GOOD and stateChange is scheduled) then
    next state is set to BAD;
    scheduleMessage(stateChange,meanBad);
else if (actual state is BAD and stateChange is scheduled) then
    next state is set to GOOD;
    scheduleMessage(stateChange,meanGood);
else
    message different from stateChange
error(Internal error);
```

2.4 Vehicle model for Plexe

The mobility model available in Veins simulator are realized in SUMO and reproduces the classical car following model [78]. In other words, the mobility model, homogeneous for each vehicle, for the cooperative driving scenario, can be described according to the ideal *acceleration-speed* profile in Fig. 2.7. This assumption means that the control effort u_{eng} (i.e. the required acceleration) and the vehicle speed are saturated to their maximum admissible value each time a vehicle accelerates or decelerates. Therefore, for every vehicle, the following maximum allowable limits have to be defined:

1. maximum acceleration a_{max};
2. minimum acceleration a_{min};
3. maximum speed v_{max}.

This implies that the traction characteristics in Veins simulator are bounded (limits on both control effort and maximum allowable speed) as:

- $a_{min} \leq u_{eng} \leq a_{max}$
- $v_{CoG} \leq v_{max}$

Until now, the mobility model, exploited in Plexe was reproduced according to the maximum vehicle acceleration and deceleration (eventually as specified by the manufacturer), inheriting this aspect from Veins. Furthermore, to simulate the drivetrain dynamics, Plexe simulator (see Eq. (2.5)) employ a first order filter [100]. The time constant, modelling the drivetrain

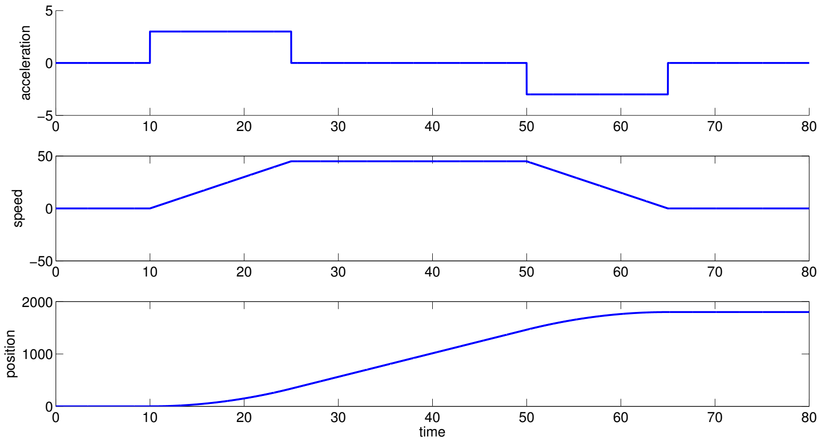


Figure 2.7: Acceleration speed and position profile. To be customized for every vehicle.

behaviour, has been set to a fixed value $\tau_{eng,d} = 0.5$ according to the approach suggested in [100]. Thus, the dynamical behaviour for each vehicle can be expressed with the following equations:

$$\begin{cases} \dot{r}_{CoG} = v_{CoG} \\ \dot{v}_{CoG} = \dfrac{1}{1 + s\tau_{eng,d}} u_{eng} \end{cases} \tag{2.5}$$

Where each variables is referred to the Center of Gravity (CoG). Starting from the model in (2.5), we propose two new solutions to embeds into Plexe simulator, that can better describe the overall dynamics of vehicles both in traction and braking phase.

2.4.1 Vehicle dynamics

As first step we extend the current model in Plexe (see Eq. (2.5)) adding friction, vehicle inertial forces and a time constant that varies according to the engine regime and gearbox status. Fundamentals equations for the vehicle dynamics are derived by using the generalized Newton's second law and the D'Alembert's principle.

The sum of the friction forces acting on a vehicle during its longitudinal motion are represented by the following equation [44, 68]:

$$F_L = F_{wind} + F_R + m_{CoG} g \sin(\theta_{road}), \tag{2.6}$$

where the terms F_L represent the total friction force acting on the vehicle. Moreover:

- F_{wind}, is the air drag [68]:

  $$F_{wind} = \frac{1}{2} c_{air} A_L \rho_a v_{CoG}^2, \tag{2.7}$$

  where c_{air} is the drag coefficient, A_L is the maximum vehicle cross section area, ρ_a is the air density.

- F_R, is the rolling resistance [128]:

  $$F_R = m_{CoG} g \left(c_{r1} + c_{r2} v_{CoG}^2 \right), \tag{2.8}$$

  where c_{r1} and c_{r2} depend for instance, on the kind of tires and tire pressure.

- $m_{CoG} g \sin(\theta_{road})$ is the gravitational force, where θ_{road} is the slope of the road.

Taking into account the Eq. (2.6), the force balance is thus:

$$F_i = \lambda m_{CoG} \dot{v}_{CoG} = F_{u_{eng}} - F_L \tag{2.9}$$

where $F_{u_{eng}}$ represent the total propelling force resulting from the control action applied at the center of gravity of the vehicle (tractive force, note that u_{eng} is the desired vehicle acceleration); $F_i = \lambda m_{CoG} \dot{v}_{CoG}$ is the inertial force acting on the center of gravity of the vehicle being λ the mass factor that takes into account the rotational inertia of the driveline. In so doing, the model is:

$$\begin{cases} \dot{r}_{CoG} = v_{CoG} \\ \dot{v}_{CoG} = \dfrac{1}{\lambda m_{CoG}} \left(F_{u_{eng}} - F_L \right) \end{cases} \tag{2.10}$$

where:

$$F_{u_{eng}} = u_{eng} \lambda m_{CoG} \tag{2.11}$$

The goal of the controller is to determine the desired acceleration u_{eng} that the engine have to impress to the vehicle so that performance requirements are meet. According to Eq. (2.11), it is possible to write Eq. (2.10) as follows:

$$\begin{cases} \dot{r}_{CoG} = v_{CoG} \\ \dot{v}_{CoG} = u_{eng} - \dfrac{1}{\lambda m_{CoG}} F_L \end{cases} \tag{2.12}$$

Now model in Eq. (2.5), can be extended as:

$$\begin{cases} \dot{r}_{CoG} = v_{CoG} \\ \dot{v}_{CoG} = \dfrac{1}{1 + \tau_{eng,d}(n)s} u_{eng} - \dfrac{1}{\lambda m_{CoG}} F_L \end{cases} \tag{2.13}$$

where the term F_L is in Eq. (2.6) and $\tau_{eng,d}$ is the minimum time delay that depends on the engine speed. It can be approximated as suggested in [68]:

$$\tau_{eng,d} = \frac{2(CYL - 1)}{CYL \cdot n} \tag{2.14}$$

where CYL is the number of cylinders and $\dfrac{1}{n}$ is the time needed for one crankshaft revolution in second. We remark that taking into account the gearbox ratio, it is possible to get $\tau_{eng,d}$ as a function of the vehicle speed. In Sec. 2.4.2 we will show how to modify Eq. (2.14) as a function of vehicle speed (see Fig. 2.13).

2.4.2 Traction/Braking force model.

Model in Eq. (2.13) can be enhanced by expressing the traction force $F_{u_{eng}}$ in case of acceleration or brake. In the first case $F_{u_{eng}}$ is the propelling force acting on the vehicle, generated by the engine, in the second case it is the braking force generated by the friction among disks and pads:

$$\begin{cases} F_{u_{eng}} = F_{brake}(t) \quad if \;\; u_{eng} \leq 0 \\ F_{u_{eng}} = T(\overline{v_{CoG}}) \quad if \;\; u_{eng} > 0 \end{cases} \tag{2.15}$$

In Sec. 2.4.3 we detail how to compute the traction effort $T(\overline{v_{CoG}})$, while in Sec. 2.5 we will give an approximate expression for the braking effort $F_{brake}(t)$.

2.4.3 The tractive effort curve $T(\overline{v_{CoG}})$: a simple procedure

The procedure to get the force-velocity diagram (*tractive effort curve*) requires engine power information. In general, the tractive effort curves are available as results of the dyno tests done by manufacturers. However it is possible to get approximately this diagram according to the following procedure.

With no loss of generality is possible to relate the speed of the vehicle v_{CoG} to the engine speed N_{eng} [101] according to the following formula:

$$v_{CoG} = \frac{N_{eng} d_{wheel} \pi}{i_d i_g} \; in \; [\mathrm{m\,min^{-1}}] \tag{2.16}$$

where i_d is the differential transmission ratio (usually it ranges from 3:1 to 6:1 for typical IC vehicle applications), i_g is the transmission gearing ratio for the $i - th$ gear (it is high for low gears and low for high gears and ranges from 5:1 to 1:1), d_{wheel} is the diameter of the tractive wheels and N_{eng} is the engine speed.

Since N_{eng} is usually expressed in Revolutions per Minute [RPM] a conversion is necessary as d_{wheel} and v_{CoG} are expressed in [m], and [$\mathrm{km\,h^{-1}}$], respectively. Thus, we obtain:

$$v_{CoG} = \frac{60 N_{eng} d_{wheel} \pi}{1000 i_d i_g} \; in \; [\mathrm{km\,h^{-1}}] \tag{2.17}$$

To relate the engine power output P_{eng} to the traction effort $F_{u_{eng}}$, we can use the following expression:

$$T(v_{CoG}) = \frac{\eta P_{eng}}{v_{CoG}} \; in \; [\mathrm{hp\,km^{-1}\,h}] \tag{2.18}$$

where P_{eng} is the power output expressed in [hp], η is the efficiency of the engine and power transmission (it ranges from 0.7 to 0.85 for most of the IC applications), v_{CoG} is the vehicle speed expressed in [$\mathrm{km\,h^{-1}}$] and T is the thrust or tractive effort produced by the engine. Before applying this equation to car, truck or bus vehicles it is necessary to multiply for a conversion factor:

$$T(v_{CoG}) = 2650 \frac{\eta P_{eng}}{v_{CoG}} \; in \; [\mathrm{N}] \tag{2.19}$$

where P_{eng} is expressed in [hp], v_{CoG} is expressed in [$\mathrm{km\,h^{-1}}$] and T is expressed in [N]. Remember that conversion factor 2650, comes out from the ratio between the positive real number 3600 and 1.34, representing respectively the conversion factor from [km h^{-1}] to [m s^{-1}] and from [kW] to [hp].

Finally, the procedure to estimate force-speed performance of a vehicle can be hence summarized by the following steps:

- compute v_{CoG} for several values of N_{eng} according to Eq. (2.17);
- substitute the corresponding velocity v_{CoG} values into Eq. (2.19) to estimate T. Since each vehicles has several feasible transmission gearing ratios, i_g it is necessary to compute several values of T for each value N_{eng} as in Eq. (2.17).

A typical tractive effort curve diagram of a vehicle equipped with an internal combustion engine, is presented in Fig. 2.8.

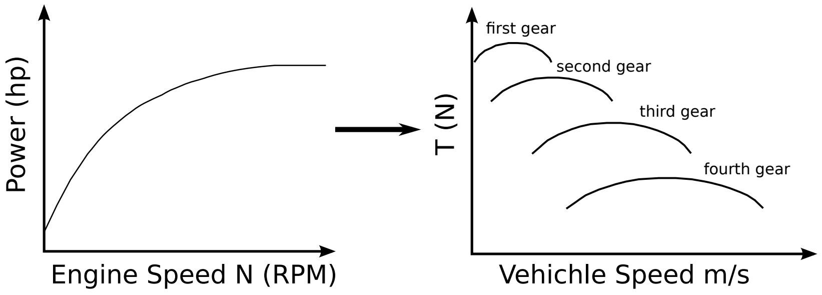


Figure 2.8: The curve $N_{eng} - P_{eng}$ is given by the manufaturer. The curve $T - v_{CoG}$ is computed with the procedure Sec. 2.4.3

To better clarify the approach, in what follow we show a numerical example. Consider as exemplar case an Alfa 147 Twin Spark 1.6 gasoline. The vehicle present a five speed gear system whose gearing ratios are the following: i_g = [3.909; 2.238; 1.52; 1.156; 0.971]. The differential ratio is constant at 3.71 : 1 and the wheel diameter is 0.4 m. The engine parameters are in Tab. 2.1[2].

Table 2.1: Engine parameters (Manufacturer) [3].

N_{eng} in [RPM]	1521	2004	2505	3012	3504	4011	4209	4517	5007	5516
P in [hp]	23.18	33.36	44.08	53.19	64.05	75.84	81.07	84.42	92.46	105.59

[2] In case power characteristics are in [kW], remember to convert the power expressed in [kW] to [hp], by multiplying for 1.34.

Table 2.2: Lookup table for the individuation of the traction effort.[4].

$\overline{v_{CoG}}$ in $\mathrm{km\,h^{-1}}$	T in N
0	5365
54.68	5365
95.51	3072
140.63	2086
184.92	1586
...	...

Now, it is possible to interpolate points in Tab. 2.1 with an opportune polynomial (see Fig. 2.9) to reconstruct the complete power characteristic curve of the engine.

Exploiting the polynomial equation and the procedure already presented for each value of the gear ratio, we get five different curves, as shown in Figs. 2.10 and 2.11:

Interpolating the maximum value of each curve in Fig. 2.10 it is possible to obtain an approximate curve representing the *tractive effort-vehicle speed* characteristic at its maximum value, as depicted in Fig. 2.11.

Starting from this diagram, according to the above considerations, it is possible to get the upper bound for the control action as:

$$F_{u_{eng}}\overline{v_{CoG}} = T\overline{v_{CoG}} \tag{2.20}$$

where $\overline{v_{CoG}}$ is the desired vehicle speed. This operation can be executed by exploiting a lookup tables or computed in real time starting from the polynomial that fit better the engine power characteristic. Results related to our case study are shown in Tab. 2.2. Values in Tab. 2.2 corresponds the traction effort between two consecutive break points computed using a linear interpolation.

Exploiting this procedure it is possible to compute the maximum acceleration as a function of the desired current speed in a simple way starting from Fig. 2.11. Now dividing the Eq. (2.20) by the vehicle mass we have:

$$u_{eng} = \frac{T}{\lambda m_{CoG}} \tag{2.21}$$

As result we obtain the *acceleration-vehicle speed* diagram in Fig. 2.12.

[3] Data are referred to Alfa Romeo 147 1.6 Twin Spark - 01 (88 kW).

[4] Data are referred to Alfa Romeo 147 1.6 Twin Spark - 01 (88 kW).

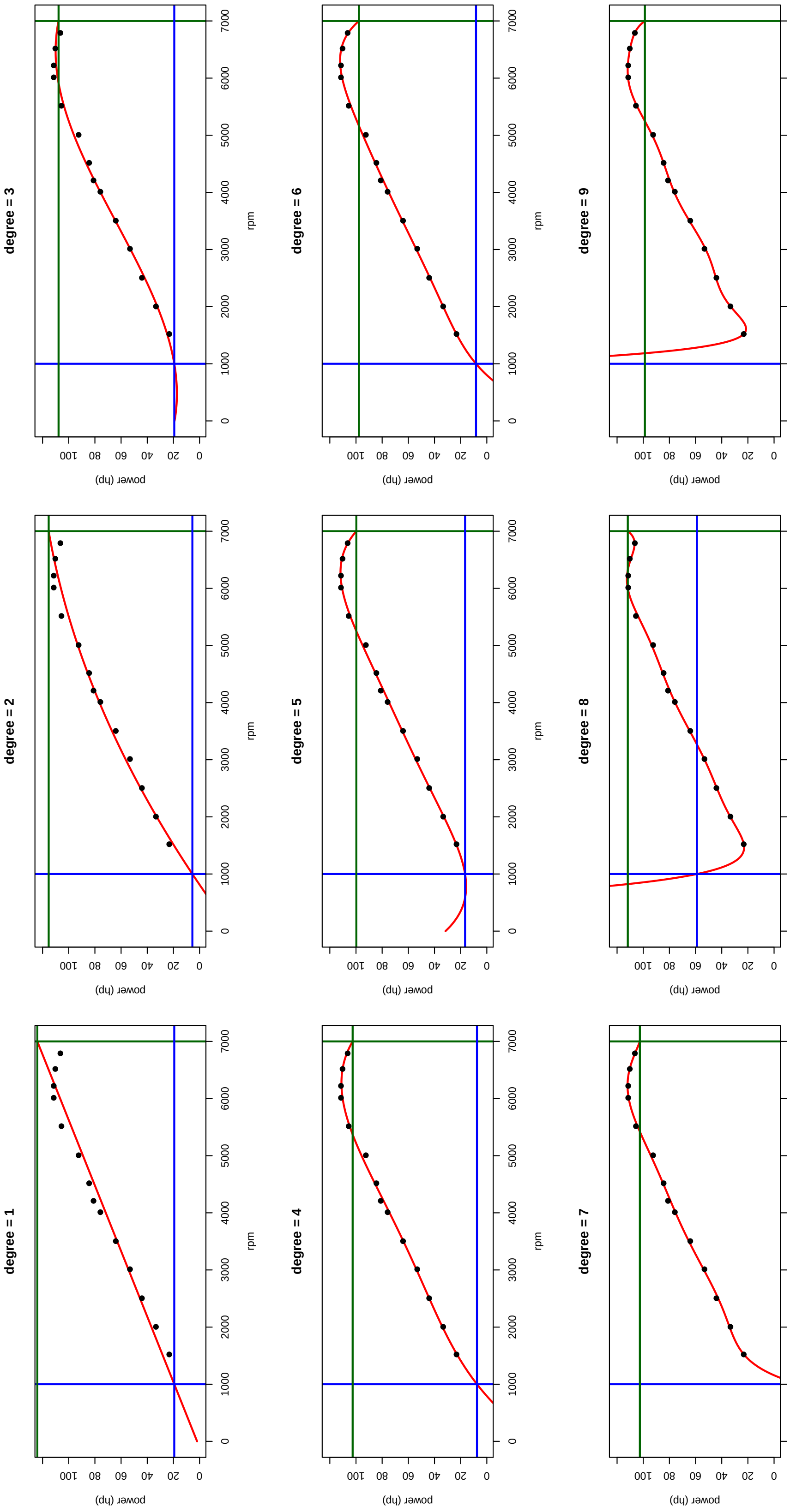


Figure 2.9: Interpolation of $N_{eng} - P_{eng}$ engine point for different grade of interpolation polinomial, with reference to Tab. 2.1

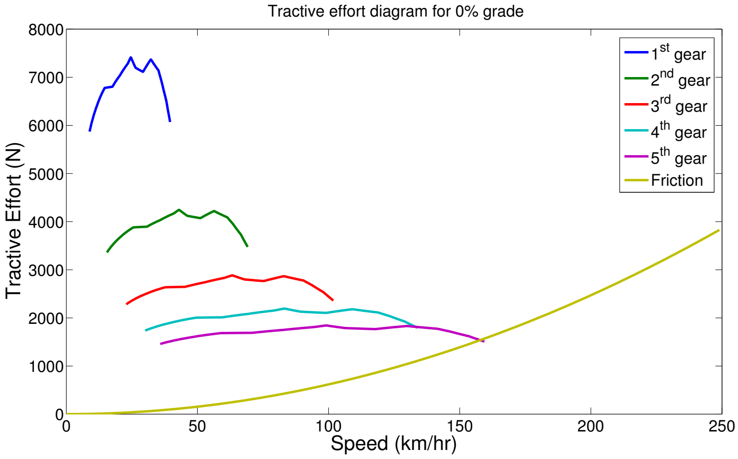


Figure 2.10: The curve T-v_{CoG} is computed with the procedure Sec. 2.4.3.

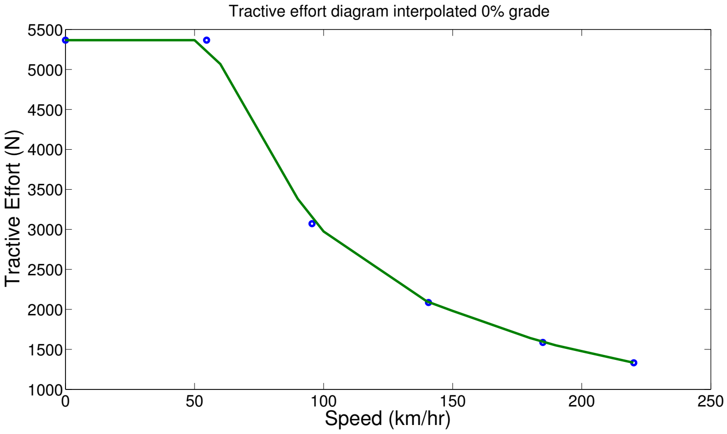


Figure 2.11: The curve T-v_{CoG} interpolated to the maximum value.

It is important to note that, due to the combustion process of the engine, the system is affected by an actuation lag. The requested torque can be only generated with a lag time of at least one cylinder segment and this lag depends on the engine speed. The minimum time lag time can be approximated as in [68]:

$$\tau_{eng,d} = \frac{2(CYL - 1)}{CYL \cdot n} \tag{2.22}$$

where CYL is the number of cylinders, $\frac{1}{n}$ is the time needed for *one crankshaft revolution*. In Fig. 2.13 the engine actuation lag time Eq. (2.14)

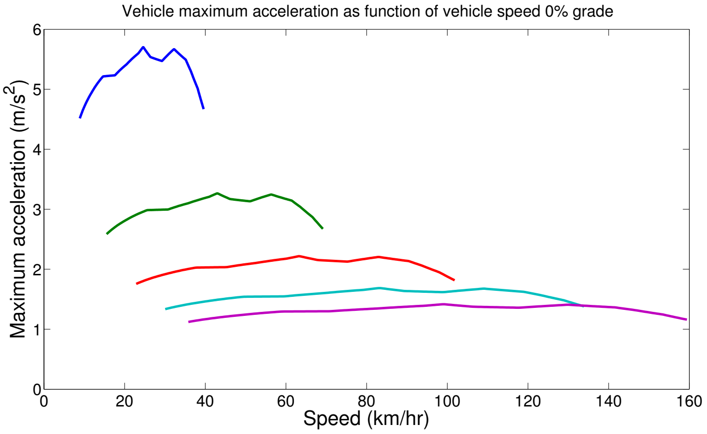


Figure 2.12: Maximum acceleration allowable as function of vehicle speed.

is represented as a function of the vehicle speed by using the conversion in Eq. (2.16).

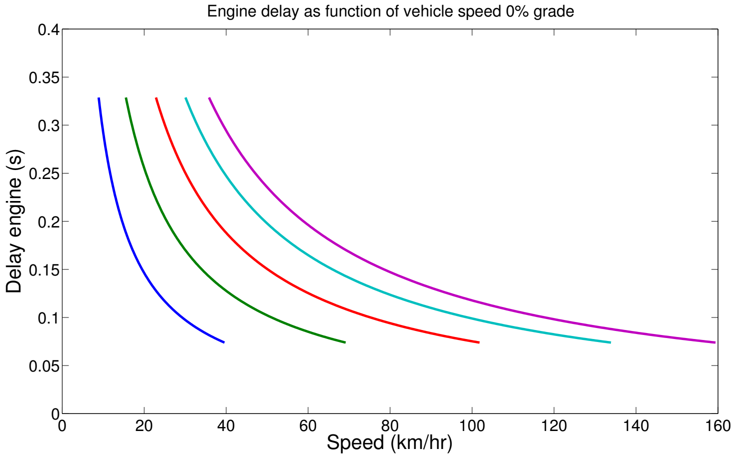


Figure 2.13: Actuation engine lag time as function of vehicle speed.

The combustion can be modelled as a delay τ_{burn}, representing the time until the opening of exhaust valve. Another delay time τ_{ex} results from time exhaust gas needs to get lambda sensor and is zero for diesel engine. In general the sum of both contributes $\tau_{de} = \tau_{ex} + \tau_{burn}$, ranges from 100 ms to 1 s. This delay will be soon embed into PLEXE to describe in approximated way the combustion phenomenon.

Before conclude this section, note that in case manufacturers give the diagram of the engine torque as a function of the engine speed, then it is possible to get the tractive effort curve by using the following expression:

$$T = \frac{M_{eng} i_g i_d \eta}{r_{wheel}} \tag{2.23}$$

2.5 Brake model for Plexe

The braking torque acting on a wheel equipped with a disk brake can be defined as:

$$T_{brake} = F_{friction} r_{eff} \tag{2.24}$$

where $F_{friction}$ is the total tangential friction force acting on the disk while r_{eff} is the effective pad radius, defined as $r_{eff} = \frac{r_{out} + r_{in}}{2}$. Eq. (2.24) can be rewritten as [28]:

$$T_{brake} = F_{friction} r_{eff} = \gamma F_{normal} r_{eff} = \gamma P_{cp} A_{ca} r_{eff} \tag{2.25}$$

where F_{normal} is the normal load on the disk surface, γ is the friction coefficient, P_{cp} is the master cylinder pressure and A_{ca} is the total actuation area of the hydraulic pistons.

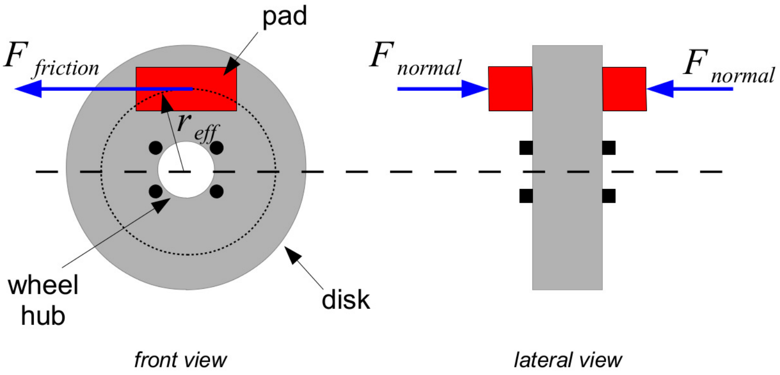


Figure 2.14: A braking system schematic

For a disk brake system there are N brake pads acting on the disks, thus the total brake torque is:

$$T_{brake} = N \gamma P_{cp} A_{ca} r_{eff} \tag{2.26}$$

Note that in what follows we will suppose γ, A_{ca} and r_{eff} constant. Thus the total amount of braking force is:

$$F_{brake} = \frac{N\gamma P_{cp} A_{ca} r_{eff}}{r_{wheel}} \tag{2.27}$$

According to Eq. (2.27) the maximum braking force is ideally reached when the master cylinder exploit its maximum pressure, $P_{cp_{max}}$ on the brake caliper i.e.: $F_{brake} = \dfrac{N\gamma P_{cp_{max}} A_{ca} r_{eff}}{r_{wheel}}$.

According to [42] it is possible to approximate the hydraulic actuation dynamics for the master cylinder with a first order differential equation, describing the pressure dynamic inside the master cylinder. Being the pressure of the master cylinder strictly tied to the braking force F_{brake} (see Eq. (2.27)), the following first order filter can be used to approximately describe the braking dynamics of the vehicle as:

$$F_{brake}(t) = \frac{1}{s\alpha + 1} F_{brake}\delta(t), \tag{2.28}$$

where the input $\delta(t)$ in Eq. (2.28) is defined as:

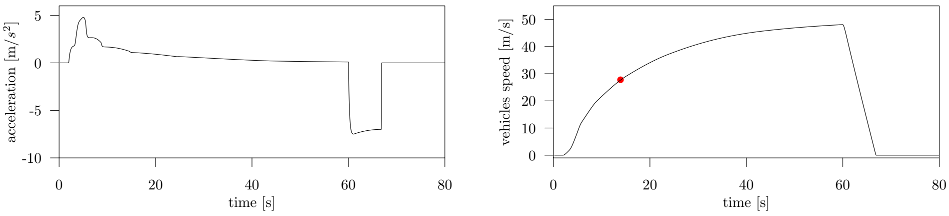
$$\begin{cases} \delta(t) = 1 \;\; if \;\; F_{u_{eng}} \leq 0 \\ \delta(t) = 0 \;\; if \;\; F_{u_{eng}} > 0. \end{cases} \tag{2.29}$$

Plexe, allows user to define the value of the braking systems parameters, in particular of α representing the braking actuation lag.

2.6 Model Parameters and Validation

According to the presented methodology, we have implemented and made available to chose in simulation, four different vehicle model: alfa-147, audi-a3, bmw-116d, mini-cooper. Data about vehicles characteristics have been retrieved at the following web site: `http://rototest-research.eu/`. All models, are customizable trough an *xml* file that gives the possibility to define/modify the value of all the parameters presented in the previous section. In Tab. 2.3 we report values of parameters that we have employed during our Plexe simulation campaign to test and validate the control strategy proposed in Chapter 3 (note that: lookup tables and curves are also available in Plexe simulation environment) [19, 29].

Every model have been validated comparing the simulated performance with them declared by manufacturer. As an exemplar case in Fig. 2.15 we show simulation results obtained for alfa 147, modelled according to what presented in Sec. 2.4.3. Results are in full agreement with what can be detected by the road tests (see `http://www.quattroruote.it/`), that assure an acceleration from $0\,\mathrm{km\,h^{-1}}$ to $100\,\mathrm{km\,h^{-1}}$ in $10.9\,\mathrm{s}$ (corresponding to the circular marker in Fig. 2.15b).



(a)

(b)

Figure 2.15: Simulated road test for Alfa 147: (a) Time history of the acceleration $a(t)$; (b) Time history of the speed $v(t)$.

Table 2.3: Vehicles transmission and power characteristics.

Audi A3 2.0 TDI	
Min engine power	11.3 kW
Max engine power	117.4 kW
Gearbox ratio	$i_1 = 3.769$ $i_2 = 2.087$ $i_3 = 1.324$ $i_4 = 0.977$ $i_5 = 0.78$ $i_6 = 0.651$
Differential ratio	$i_d = 3.45$
Car mass	1400 kg
Maximum car section	2.08 m^2
Engine efficiency	0.9
BMW 1 Series 116d 2.0	
Min engine power	15.7 kW
Max engine power	84.2 kW
Gearbox ratio	$i_1 = 4.002$ $i_2 = 2.108$ $i_3 = 1.38$ $i_4 = 1.0$ $i_5 = 0.78$ $i_6 = 0.645$
Differential ratio	$i_d = 3.071$
Car mass	1295 kg
Maximum car section	2.09 m^2
Transmission efficiency η	0.95
Mini Cooper 1.6 2002	
Min engine power	16.3 kW
Max engine power	80.2 kW
Gearbox ratio	$i_1 = 3.417$ $i_2 = 1.947$ $i_3 = 1.333$ $i_4 = 1.051$ $i_5 = 0.848$
Differential ratio	$i_d = 3.94$
Car mass	1150 kg
Maximum car section	1.98 m^2
Transmission efficiency η	0.95
Alfa Romeo 147 1.6 Twin Spark	
Min engine power	17.3 kW
Max engine power	83.2 kW
Gearbox ratio	$i_1 = 3.909$ $i_2 = 2.238$ $i_3 = 1.520$ $i_4 = 1.156$ $i_5 = 0.971$
Differential ratio	$i_d = 3.714$
Car mass	1400 kg
Maximum car section	2.1 m^2
Transmission efficiency η	0.9
Common parameters	
Rolling coefficient c_{r1}	0.0136
Rolling coefficient c_{r2}	$5.18 \cdot 10^{-7}$
Air friction coefficients c_{air}	0.31
Mass factor λ	1.089
Wheels diameter	0.62 m
Air density ρ_a	1.2 $\mathrm{kg\,m}^{-3}$
Maximum vehicle acceleration a_{max}	$0.25 \times g$
Minimum vehicle deceleration a_{min}	$0.5 \times g$
Typical braking system parameters	
Pad-disk friction coefficient γ	0.4
Typical maximum calliper pressure$P_{cp_{max}}$	5 MPa to 10 MPa
Actuation area of the hydraulic piston(s) A_{ca}	1810 mm^2
External disk radius r_{out}	110 mm
Internal disk radius r_{in}	75 mm

Chapter 3

Cooperative driving algorithm

Abstract

A key element of cooperative driving systems for the longitudinal control of vehicles is the design of algorithms able to impose a common motion to the ensemble by acting at a single vehicle level on the base of neighbours behaviour. The aim is to regulate speed and relative position of each vehicle in the platoon to that of the respective predecessor and of the leading vehicle (usually the first vehicle in the string) according to a required spacing policy. Following the paradigm of dynamical networks, the connected vehicle system can be represented as a complex delayed delayed network, in which the delay model the transmission time among vehicles. In this chapter we present two distributed control algorithm for cooperative driving system. The first solves a third-order consensus problem, while the latter solves an adaptive synchronization problem in a vehicular network in the presence of time-varying heterogeneous delays.

3.1 Introduction

A key element of cooperative driving systems for the longitudinal control of vehicles fleet is the design of algorithms able to impose a common motion to the ensemble by acting at a single vehicle level on the base of neighbours behaviour. The aim is to regulate speed and relative position of each vehicle in the platoon to that of the respective predecessor and of the leading vehicle (usually the first vehicle in the string) according to a required spacing policy. As information on the surrounding vehicles can be obtained both via sensors and/or via V2V wireless communication, different time-varying delays are associated to the different types of interconnections to account for the different devices/communication tools used to gather information. To guarantee platooning in the presence of time-varying delays, a novel control algorithm based on distributed consensus has been recently proposed from authors in [31,32]. Note that platoon maintenance and its string stability (i.e robustness in the presence of periodic disturbance acting on the leader motion) may be compromised by the presence of time varying delays affecting the information received via wireless communication [76]. Specifically, following the paradigm of dynamical networks, the connected vehicle system in the presence of delays is represented as a network where each node is a vehicle characterized by its own dynamics; the presence/absence of edges mimics the presence/absence of interconnections among vehicles; the structure of vehicles communication is encoded in the network topology; communication delays may be associated to links (depending on the specific link features). In so doing the designed algorithm are able to cope with different platoon topologies originated from the specific communication infrastructure, that hence are not restricted to common pairwise interactions (e.g. predecessor-following [100]) as well as with the integration of both sensor-based and communication-based vehicle technologies.

In what follow we will present two approach specifically designed to take into account spacing policy constraints, communication logical topology, as well as impairments as time-varying delays. The approaches allows the integration of both sensor-based and communication-based vehicle technologies and, if the delay affecting information gathered via on-board sensors (like, e.g., radars) is smaller than the wireless communication delay, it might be considered negligible. In Sec. 3.2 we present a distributed control algorithm for cooperative driving system that solves a third-order consensus problem, while in Sec. 3.3 we present an algorithm that solves an adaptive synchronization problem in a vehicular network in the presence of time-varying heterogeneous delays. In consensus problem, we assume that the goal is to make that every vehicle have to reach the same velocity of the leader vehicle (set-point problem). In synchronization problem, the target is to synchronize the dynamics of all agent of the platoon to the leader dynamics (tracking-problem). The problem essentially consists in leader tracking manoeuvres, whose velocity is not constant as in consensus problem. In both strategies, vehicles within the platoon are supposed to be equipped with on-board sensors (measuring relative and absolute position, speed and acceleration) as well as wireless V2V-DSRC/WAVE [2, 3] access network with beaconing messages to share information with neighbours and to receive reference signals.

3.1.1 Network model for Inter-Vehicular Communication

To model the dynamical network we exploit the complex network theory (see Sec. A.3). The inter-vehicle communication structure can be modelled by a graph where every vehicle is a node. Hence, a platoon of N vehicles is represented as a directed graph $\mathcal{G}$ (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order N characterized by the set of nodes $\mathcal{V} = \{1, \ldots, N\}$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The topology of the graph is associated to an adjacency matrix with non-negative elements $\mathcal{A} = [\alpha_{ij}]_{N \times N}$. In what follows we assume $\alpha_{ij} = 1$ in the presence of a communication link from node j to node i, otherwise $\alpha_{ij} = 0$. Moreover, $\alpha_{ii} = 0$ (i.e., self-edges (i, i) are not allowed unless otherwise indicated). The presence of edge (i, j), $(i, j) \in \mathcal{E}$, means that vehicle i can obtain information from vehicle j, but not necessarily *vice versa*, according to the classical definition used for example in [94]). Note that the existence of edge (i, j) means that i uses the information received by j and not only that i is within the communication range of j. Arrows indicating the direction of the information flow among vehicles and arrows indicating the direction of edges in the associated network graphs have opposite directions since we adopted for the network edges the definition used in the network literature (e.g. [94, 138]). Note that, defining the degree matrix as $\Delta = \text{diag}\{\Delta_1, \Delta_2, ..., \Delta_N\}$, with $\Delta_i = \sum_{j \in \mathcal{V}} \alpha_{ij}$, the Laplacian of the directed graph $\mathcal{G}$ can be defined as $L = \Delta - \mathcal{A}$. We say that j is *reachable* from i if there exists a path from node i to node j. In the rest of the chapter we consider N vehicles together with a leader vehicle taken as an additional agent labelled with the index zero, i.e., node 0. We use an augmented weighted directed graph $\mathcal{G}_{N+1}$ to model the resulting network topology. We assume node 0 is *globally reachable* in $\mathcal{G}_{N+1}$ if there is a path in $\mathcal{G}_{N+1}$ from every node i in $\mathcal{G}_N$ to node 0 [60]. This inter-vehicle communication model will be exploited to model either the consensus problem, either the synchronization problem.

3.2 Platooning as a Consensus Problem

In this section we present a distributed control algorithm able to guarantee the third-order consensus in the vehicular network in the presence of time-varying heterogeneous delays. The proposed approach enhances the cooperative driving algorithms presented in [31,32] since: *(i)* vehicles accelerations signals are embedded into the control action as to improve the control reactivity during rapid leader acceleration manoeuvres and avoid falling too far behind the vehicle ahead [100]; *(ii)* vehicles dynamics are described by a third-order drivetrain linear model well established in the automotive literature (e.g. see [100]) rather than being simply approximated as second-order inertial agents; *(iii)* new necessary and sufficient conditions to analytically guarantee consensus for time-delayed system are derived by finding a proper Krasovskii functional. Note that results on high-order consensus in the presence of heterogeneous time varying delays (also denominated non-uniform or multiple) are mainly restricted to first and second order systems [138] and often the different delays are also approximated at node level with a unique time-varying delay [63]. Only recent attempts address the problem for third- or n-order agents, but in the absence of a leader and for generic integrators networks [26]. Usually consensus in delayed networks is solved under the hypothesis of uniform (homogenous) delays [61]. Fewer attempts have been proposed in the technical literature to guaranteed high-order consensus in the presence of heterogeneous time varying delays (also denominated non-uniform or multiple) and results are mainly restricted to first and second order systems (e.g. [23, 138]). Note that often the different delays are also approximated at node level with a unique time-varying delay for each node. At the best of our knowledge, only very recent results investigate consensus for third or n-th order systems in the presence of heterogeneous time-varying delays [26]. In [26] the problem is addresses in the different case of a leaderless network of identical integrators by exploiting a different coupling protocol structure that implements a proportional action on only position errors modulated by a unique control gain. Note that the exploitation of different control gains, as proposed in the what follow, is a degree of freedom that allows designers to tune the control action independently for each agent or, as in our case, vehicles.

Consider a group of N vehicles ($i = 1, \dots, N$) and a leader moving along a single lane. The longitudinal dynamics of each vehicle in the platoon is mainly governed by its drivetrain dynamic. Although complexity of drivetrain models is well known in literature, here we use the following approximated drivetrain dynamics [80, 100]:

$$\begin{aligned} &\dot{r}_i = v_i(t); \\ &\dot{v}_i = a_i(t); \\ &\dot{a}_i = -\frac{1}{T_i} a_i(t) + \frac{1}{T_i} u_i(t) \end{aligned} \tag{3.1}$$

where r_i [m] and v_i [m/s] and a_i are the i-th vehicle position, velocity and acceleration, measured with respect to a road reference frame, T_i [s] is the time constant of the drivetrain, and u_i denotes the control signal to be appropriately chosen to achieve the desired position, maintain a desired speed and perform braking manoeuvres. We assume that a lower level control exists on each vehicle delivering the demanded reference u_i [100]. Typically, the time constant of the drivetrain $T_i > 0$ depends upon specific vehicle features. Assuming that the leader moves with a constant desired velocity, i.e. $\dot{r}_0(t) = v_0$, $\dot{v}_0 = a_0 = 0$, the platooning control goal can be expressed as the following third-order network consensus problem:

$$r_i(t) \to r_0(t) + d_{i0};\ v_i(t) \to v_0;\ a_i(t) \to 0 \tag{3.2}$$

where $d_{i0} = h_{i0} v_0 + d_{i0}^{st}$ is the desired distance of vehicle i from the leader that can be set according to the desired spacing policy [32], [31] [91]. The consensus goal in (3.2) is achieved here by using an appropriate distributed strategy, depending form local state variables as well as from the information received by the neighbouring vehicles (e.g., within the transmission range), that embeds spacing policy requirements:

$$\begin{aligned} u_i \quad &= -b\left[v_i(t) - v_0(t)\right] - \gamma\left[a_i(t) - a_0(t)\right] + \\ &+ \frac{1}{\Delta_i} \sum_{j=0}^{N} k_{ij} \alpha_{ij} \left(\tau_{ij}(t) v_0\right) - \\ &- \frac{1}{\Delta_i} \sum_{j=0}^{N} k_{ij} \alpha_{ij} \left(r_i(t) - r_j(t - \tau_{ij}(t)) - h_{ij} v_0(t) - d_{ij}^{st}\right) \end{aligned} \tag{3.3}$$

where α_{ij} models the network topology emerging from the presence/absence of a communication link between vehicles i and j (we assume $\alpha_{0j} = 0$, $\forall j = 0, \dots, N$, since the leader does not receive data from any other vehicle);

$\Delta_i = \sum_{j=0}^{N} \alpha_{ij}$ is the degree of vehicle/agent i, i.e., the number of vehicles establishing a communication link with vehicle i; control parameters $k_{ij} > 0$, $b > 0$ and $\gamma > 0$ act as stiffness and damping gains to be appropriately tuned to regulate the mutual behaviour among neighbour vehicles; $\tau_{ij}(t)$ and $\tau_{i0}(t)$ are the time-varying delays affecting the communication with the i-th vehicle when information is transmitted from vehicle j and from the leader, respectively (in general $\tau_{ij}(t) \neq \tau_{ji}(t)$); h_{ij} is the *constant time headway* (i.e., the time necessary for vehicle $i-$th to travel the distance separating it from its predecessor); d_{ij}^{st} is the desired distance between vehicles i and j at standstill. Delays $\tau_{ij}(t)$ are assumed to be bounded and detectable. In particular, the communication delay over a link can be evaluated by each vehicle when information is received, since each vehicle transmits information with a timestamp (i.e., the time instant when the information is sent) [23]. Note that the delay depends on the different devices used to gather information at each specific link. Hence, when a vehicle is equipped with on-board sensors (like, e.g., radars), the delay affecting the state measurements of the preceding vehicle might be considerably smaller than the wireless communication delay, so for that link the approximation $\tau_{ij}(t) = 0$ might be used.

3.2.1 Delayed vehicular network

To prove asymptotic stability of the closed-loop dynamics driven by the control action (3.3), we first define position and velocity and accelerations errors with respect to the reference signals $r_0(t), v_0, a_0$ $(i = 1, \ldots, N)$ as:

$$\begin{aligned} \bar{r}_i &= (r_i(t) - r_0(t) - h_{i0}v_0 - d_{i0}^{st}); \\ \bar{v}_i &= (v_i(t) - v_0), \\ \bar{a}_i &= (a_i(t) - a_0) = a_i(t). \end{aligned} \tag{3.4}$$

Re-writing the coupling control action u_i in terms of the state errors $\bar{r}_i$ and $\bar{v}_i$ and expressing headway constants h_{ij} and standstill distances d_{ij}^{st} with respect to the leading vehicle, namely $h_{ij} = h_{i0} - h_{j0}$ and $d_{ij}^{st} = d_{i0}^{st} - d_{j0}^{st}$, after some algebraic manipulation, the closed-loop dynamics for the generic

i-th platoon vehicle can be rewritten as:

$$\begin{cases} \dot{\bar{r}}_i & = & \bar{v}_i, \\ \dot{\bar{v}}_i & = & \bar{a}_i, \\ \dot{\bar{a}}_i & = & \frac{1}{T_i}\left[-(k_{i0}\alpha_{i0} + \sum_{j=1}^{N} k_{ij}\alpha_{ij})\bar{r}_i(t) - b\bar{v}_i\,(t) + \right. \\ & & \left. (-1-\gamma)\,a_i(t) + \frac{1}{\Delta_i}\sum_{j=1}^{N} k_{ij}\alpha_{ij}\left[\bar{r}_j\left(t-\tau_{ij}\left(t\right)\right)\right] \right]. \end{cases} \tag{3.5}$$

To describe the platoon dynamics in the presence of the different time-varying delays associated to the different links in a more compact form, we define here the position, speed and acceleration error vectors as $\bar{r} = [\bar{r}_1, \ldots, \bar{r}_i \ldots, \bar{r}_N]^\top$, $\bar{v} = [\bar{v}_1, \ldots, \bar{v}_i \ldots, \bar{v}_N]^\top$, $\bar{a} = [\bar{a}_1, \ldots, \bar{a}_i \ldots, \bar{a}_N]^\top$; the error state vector as $\bar{x}\,(t) = \left[\begin{array}{c} \bar{r}^\top\,(t)\;\bar{v}^\top\,(t)\;\bar{a}^\top\,(t) \end{array} \right]^\top$ and we furthermore define $\tau_p(t)$, $p = 1, 2, ..., m$ (with $m \leq N(N-1)$) as an element of the following sequence of time-delays $\{\tau_{ij}(t) : i, j = 1, 2, ..., N, i \neq j)\}$. Note that m is equal to its maximum, $N(N-1)$, if the platoon topology is represented by a directed complete graph and all time-delays are different.

According to the above definitions following the approach in [32], the closed-loop vehicular network can be represented as the following set of time-delayed differential equations:

$$\dot{\bar{x}}\,(t) = A_0\bar{x}\,(t) + \sum_{p=1}^{m} A_p\bar{x}\,(t-\tau_p\,(t))\,, \tag{3.6}$$

where

$$A_0 = \left[\begin{array}{ccc} 0_{N\times N} & I_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & I_{N\times N} \\ -T\widetilde{K} & -T\widetilde{B} & -\widetilde{T} \end{array} \right], \tag{3.7}$$

$$A_p = \left[\begin{array}{ccc} 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ T\widetilde{K}_p & 0_{N\times N} & 0_{N\times N} \end{array} \right], \tag{3.8}$$

with

$$T = \mathrm{diag}\left\{\frac{1}{T_1}, \ldots, \frac{1}{T_N}\right\} \in \mathbb{R}^{N\times N}; \tag{3.9}$$

$$\widetilde{T} = (1+\gamma)T \in \mathbb{R}^{N\times N}; \tag{3.10}$$

$$\widetilde{B} = \mathrm{diag}\{b, \ldots, b\} \in \mathbb{R}^{N\times N}; \tag{3.11}$$

$$\widetilde{K} = \mathrm{diag}\left\{\tilde{k}_{11}, \ldots, \tilde{k}_{NN}\right\} \in \mathbb{R}^{N\times N} \;\; being\tilde{k}_{ii} = \frac{1}{\Delta_i}\sum_{j=0}^{N} k_{ij}\alpha_{ij}; \tag{3.12}$$

and $\tilde{K}_p = [\bar{k}_{pij}] \in \mathbb{R}^{N\times N}$ $(p = 1, \ldots, m)$ is the matrix defined according to [31] as:

$$\bar{k}_{pij} = \begin{cases} \dfrac{\alpha_{ij}k_{ij}}{\Delta_i}, & j \neq i, \tau_p(\cdot) = \tau_{ij}(\cdot), \\ 0, & j \neq i, \tau_p(\cdot) \neq \tau_{ij}(\cdot). \\ 0, & j = i. \end{cases} \tag{3.13}$$

3.2.2 Stability Analysis

The stability proof is based on recasting the delayed closed-loop dynamics as a set of functional differential equations for which it is possible to find a quadratic Lyapunov-Krasovskii function [46] [49]. Note that the Lyapunov-Krasovskii approach has shown to be less conservative with respect to similar approaches for the analysis of delayed systems [47]. From the Leibniz-Newton formula it is known that [109]:

$$\bar{x}\left(t - \tau_p(t)\right) = \bar{x}\left(t\right) - \int_{-\tau_p(t)}^{0} \dot{\bar{x}}\left(t+s\right) ds. \tag{3.14}$$

Hence, substituting (3.6) into (3.14) we have:

$$\bar{x}\left(t - \tau_p(t)\right) = \bar{x}\left(t\right) - \sum_{q=0}^{m} A_q \int_{-\tau_p(t)}^{0} \bar{x}\left(t+s-\tau_q\left(t+s\right)\right) ds, \tag{3.15}$$

where matrices $A_0, A_1, \ldots, A_m$ are defined in (3.7) and (3.8) and $\tau_0\left(t+s\right) \equiv 0$. Using the above transformation, the time-delayed system in (3.6) can be transformed into:

$$\begin{aligned} \dot{\bar{x}}\left(t\right) &= A_0\bar{x}\left(t\right) + \sum_{p=1}^{m} A_p\bar{x}\left(t\right) + \\ &- \sum_{p=1}^{m}\sum_{q=0}^{m} A_pA_q \int_{-\tau_p(t)}^{0} \bar{x}\left(t+s-\tau_q\left(t+s\right)\right) ds. \end{aligned} \tag{3.16}$$

From (3.7) and (3.8) it follows that $A_pA_q = 0$ when $p = 1, \ldots m$ and $q = 1, \ldots, m$ $(q \neq 0)$. Hence, the system defined in (3.6) can be rewritten as:

$$\dot{\bar{x}}\left(t\right) = F\bar{x}\left(t\right) - \sum_{p=1}^{m} C_p \int_{-\tau_p(t)}^{0} \bar{x}\left(t+s\right) ds \tag{3.17}$$

where

$$C_p = A_pA_0 = \begin{bmatrix} 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & T\widetilde{K}_p & 0_{N\times N} \end{bmatrix}, \tag{3.18}$$

and

$$F = A_0 + \sum_{p=1}^{m} A_p = \begin{bmatrix} 0_{N\times N} & I_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & I_{N\times N} \\ -T\widehat{K} & -T\widetilde{B} & -\widetilde{T} \end{bmatrix}, \tag{3.19}$$

with

$$\widehat{K} = -\sum_{p=1}^{m} \widetilde{K}_p + \widetilde{K}. \tag{3.20}$$

Furthermore, the following Lemmas hold.

Lemma 1. *[32] Supposing $k_i = \dfrac{k_{i0}\alpha_{i0}}{\Delta_i} \geq 0$ $(i = 1, \ldots, N)$, the matrix $\widehat{K}$ in (3.20) is positive stable if and only if node 0 is globally reachable in $\mathcal{G}_{N+1}$.*

Note that from Lemma 1, it follows that the matrix $\widehat{K}_T$ defined as

$$\widehat{K}_T = T\widehat{K}, \tag{3.21}$$

is also positive stable since $T > 0$ (see (3.9)).

Lemma 2. *Consider the matrix F defined in (3.19). F is Hurwitz stable if and only if $\widehat{K}_T$ in (3.21) is a positive stable matrix with control gains $\gamma > 0$ and $b > 0$ in (3.3) so that the following inequalities holds:*

$$\begin{aligned} \left(\frac{b}{T_{\max}}\right)\left(\frac{1+\gamma}{T_{\max}}\right) - \mu_R &> 0, \\ \mu_R\left[\mu_R - \left(\frac{1+\gamma}{T_{\max}}\right)\left(\frac{b}{T_{\max}}\right)\right]^2 - \left(\frac{1+\gamma}{T_{\max}}\right)^3 \mu_I^2 &> 0, \end{aligned} \tag{3.22}$$

where μ_i is the i-th eigenvalue of $\widehat{K}_T$, $\mu_R = \max_i \{Re(\mu_i)\}$ and $\mu_I = \max_i \{Im(\mu_i)\}$ $(i = 1, \ldots, N)$, $T_{\max} = \max_i \{T_i\}$ the maximum drivetrain constant.

Proof. (*Sufficiency*) Exploiting Shur's formula [60], the characteristic polynomial of F can be computed as:

$$\begin{aligned} &\det\left(\lambda I_{3N\times 3N} - F\right) = \\ &= \det\left(\begin{bmatrix} \lambda I_{N\times N} & -I_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \lambda I_{N\times N} & -I_{N\times N} \\ \widehat{K}_T & T\widetilde{B} & \lambda I_{N\times N} + \widetilde{T} \end{bmatrix}\right) = \\ &= \det\left(\lambda^3 I_{N\times N} + \lambda^2 \widetilde{T} + \lambda T\widetilde{B} + \widehat{K}_T\right) = \\ &= \prod_{i=1}^{N}\left[\lambda_i^3 + \left(\frac{1+\gamma}{T_i}\right)\lambda_i^2 + \left(\frac{b}{T_i}\right)\lambda_i + \mu_i\right] = \prod_{i=1}^{N} \pi_i(\lambda_i), \end{aligned} \tag{3.23}$$

Choosing now the control gains $\gamma > 0$ and $b > 0$ so that conditions in (3.22) are fulfilled, we have that the "north-westerly" minors of the Bilharz matrix [38] associated to the i-th polynomial $\pi_i(\lambda_i)$ in (3.23), i.e.

$$\begin{aligned}
D_{1_i} &= \frac{1+\gamma}{T_i}, \\
D_{2_i} &= \left(\frac{1+\gamma}{T_i}\right)^2 \frac{b}{T_i} - Re(\mu_i)\left(\frac{1+\gamma}{T_i}\right), \\
D_{3_i} &= Re(\mu_i)\left[Re(\mu_i) - \left(\frac{1+\gamma}{T_i}\right)\left(\frac{b}{T_i}\right)\right]^2 - \left(\frac{1+\gamma}{T_i}\right)^3 (Im(\mu_i))^2,
\end{aligned} \tag{3.24}$$

are positive definite $\forall i$. Exploiting Lemma 5, it can be deduced that roots of $\pi_i(\lambda_i)$ are in the open left half of Argand-Gauss plane $\forall i$. Thus, F is Hurwitz stable since all the eigenvalues of F have negative real parts.

(*Necessity*) Assume, for the sake of contradiction, that the statement does not hold, hence F is Hurwitz stable and $\widehat{K}_T$ is negative definite. Therefore, the determinants D_{3_i} defined in (3.24) will be never positive definite and, hence, according to Lemma 5, it follows that that matrix F in (3.23) is Hurwitz unstable. This leads to a contradiction since F is Hurwitz stable for hypothesis so proving the claim. □

Consensus in the presence of the heterogeneous time-varying delays can be now guaranteed under the classical constraints on bounded delay functions [40, 46, 105], *i.e.* $\tau_p(t) \in [0, \tau_{\max}]$, $\dot{\tau}_p(t) \in (-\infty, d_p]$ ($\forall t$ $\forall p$) and $d_p \leq 1$ being $\mu = \max_p d_p$, according to the following Theorem.

Theorem 1. *Consider the time-delayed system in (3.6). Set the control parameters k_{ij}, b and γ in (3.3) as in Lemmas 1 and 2. Assume delays $\tau_p(t)$ ($p = 1, \ldots, m$) to be bounded. Then there exists a constant $\tau^\star > 0$ such that, for $\tau_p(t) < \tau^\star$ $\forall p\, \forall t$, we have*

$$\lim_{t\to\infty} \bar{x}(t) = 0,$$

if and only if node 0 is globally reachable in $\mathcal{G}_{N+1}$.

Proof. (*Sufficiency*): Consider the following Krasovskii functional for system in (3.17):

$$V(\bar{x}(t)) = \bar{x}(t)^\top P\bar{x}(t) + \sum_{p=1}^{m}\int_{t-\tau_p(t)}^{t} \bar{x}(\xi)^\top S_p\bar{x}(\xi)d\xi, \tag{3.25}$$

being $P = P^\top > 0$ and $S_p > 0$ ($p = 1, .., m$) appropriately chosen matrices, and define the following continuous non-decreasing and positive functions, satisfying the hypotheses of Theorem 7, as:

$$\begin{aligned}
u(\bar{x}(t)) &= \bar{x}(t)^\top P\bar{x}(t), \\
v(\bar{x}(t-\tau^\star)) &= \bar{x}(t)^\top P\bar{x}(t) + \sum_{p=1}^{m}\int_{t-\tau^\star}^{t} \bar{x}(\xi)^\top S_p\bar{x}(\xi)d\xi,
\end{aligned} \tag{3.26}$$

where $\tau^\star$ is the maximum admissible delay.

From definitions in (3.25) and (3.26) we have that condition (A.11) in Theorem 7 is fullfilled, i.e.

$$u(\bar{x}(t)) \leq V(\bar{x}(t)) \leq w(\bar{x}(t-\tau^\star)). \tag{3.27}$$

Now differentiating Krasovskii functional in (3.25) yields:

$$\dot{V}(\bar{x}) = \dot{\bar{x}}(t)^\top P\bar{x}(t) + \bar{x}(t)^\top P\dot{\bar{x}}(t) + \sum_{p=1}^{m} \left(\bar{x}(\xi)^\top S_p \bar{x}(\xi)\dot{\xi}\right)\Bigg|_{\xi=t-\tau_p(t)}^{\xi=t} . \tag{3.28}$$

Substituting (3.17) into (3.28), after some algebraic algebraic manipulations, it follows:

$$\begin{aligned} &\dot{V}(\bar{x}) = \bar{x}(t)^\top \left(PF + F^\top P + \sum_{p=1}^{m} S_p\right)\bar{x}(t) + \\ &-2\bar{x}(t)^\top P \sum_{p=1}^{m} C_p \int_{-\tau_p(t)}^{0} \bar{x}(t+s)ds + \\ &-\sum_{p=1}^{m}\left(\bar{x}(t-\tau_p(t))^\top S_p \bar{x}(t-\tau_p(t)\right)(1-\dot{\tau_p}). \end{aligned} \tag{3.29}$$

Since node 0 is globally reachable in $\mathcal{G}_{N+1}$, the matrix $\widehat{K}$ is positive stable by selecting gains k_i according to Lemma 1. Now we are under the hypotheses of Lemma 2, hence, setting b and γ as in (3.22) it follows that matrix F in (3.19) is Hurwitz stable and we have $PF + F^T P = -Q$ with $Q > 0$ and $P > 0,\ P = P^\top$ according to Lyapunov theory.

Furthermore, for any generic positive definite matrix Ξ it holds [60]: $2a^\top c \leq a^\top \Xi a + c^\top \Xi^{-1} c$. Therefore, integrating both sides of this inequality and setting $a^\top = \overline{x}^\top PC_p$, $c = \overline{x}(t+s)$, $\Xi = P^{-1}$, equality (3.29) becomes:

$$\begin{aligned} &\dot{V}(\bar{x}) \leq -\bar{x}(t)^\top Q\bar{x}(t) + \bar{x}(t)^\top \sum_{p=1}^{m} S_p \bar{x}(t) + \\ &-\sum_{p=1}^{m}\left[\tau_p(t)\bar{x}(t)^\top PC_pP^{-1}C_p^\top P^\top \bar{x}(t) + \int_{-\tau_p(t)}^{0} \bar{x}(t+s)^\top P\bar{x}(t+s)ds\right] \\ &-\sum_{p=1}^{m}\left(\bar{x}(t-\tau_p(t))^\top S_p\bar{x}(t-\tau_p(t))\right)(1-\dot{\tau_p}(t)). \end{aligned} \tag{3.30}$$

Exploiting the Hadamard Inequality for the integral term (see Lemma 6) and the bound on the delay functions [40], inequality (3.30) can be recast as

$$
\begin{aligned}
&\dot{V}(\bar{x}) < -\bar{x}(t)^\top Q\bar{x}(t) + \bar{x}(t)^\top \sum_{p=1}^{m} S_p\bar{x}(t)\\
&-\sum_{p=1}^{m}\Bigg[\tau^\star \bar{x}(t)^\top PC_pP^{-1}C_p^\top P^\top \bar{x}(t) + \\
&+\frac{\tau^\star}{2}\left(\bar{x}(t)^\top P\bar{x}(t) + \bar{x}(t-\tau_p(t))^\top P\bar{x}(t-\tau_p(t))\right)\Bigg] + \\
&-\sum_{p=1}^{m}\left(\bar{x}(t-\tau_p(t))^\top S_p\bar{x}(t-\tau_p(t))\right)(1-d_p).
\end{aligned}
\tag{3.31}
$$

Defining now an augmented state error vector as $\xi(t) = [\bar{x}(t), \bar{x}(t-\tau_1(t)), \ldots, \bar{x}(t-\tau_m(t))]^\top$, it is possible to write (3.31) in a more compact form as:

$$
\dot{V}(t) < \xi^\top(t)\Lambda\xi(t), \tag{3.32}
$$

being

$$
\Lambda = \begin{bmatrix} \Lambda_1 & 0_{N\times N} & \cdots & 0_{N\times N} \\ \star & \Lambda_2 & 0_{N\times N} & \vdots \\ \star & \star & \ddots & 0_{N\times N} \\ \star & \cdots & \star & \Lambda_{m+1} \end{bmatrix}, \tag{3.33}
$$

with

$$
\begin{aligned}
&\Lambda_1 = -Q + \sum_{p=1}^{m} S_p - \sum_{p=1}^{m}\frac{\tau^\star}{2}\left[2PC_pP^{-1}C_p^\top P^\top + P\right],\\
&\Lambda_2 = -\frac{\tau^\star}{2}P - S_1(1-d_1),\\
&\Lambda_3 = -\frac{\tau^\star}{2}P - S_2(1-d_2),\\
&\vdots\\
&\Lambda_{m+1} = -\frac{\tau^\star}{2}P - S_m(1-d_m).
\end{aligned}
\tag{3.34}
$$

Now, according to Lyapunov-Krasovskii Theorem 7 (see condition (A.12)), to guarantee uniform stability of the time-delayed closed-loop system, it is sufficient to note that matrix Λ in (3.33) is negative definite when

$$
\tau^\star < \frac{\left\|\sum_{p=1}^{m} S_p - Q\right\|}{\left\|\sum_{p=1}^{m} PC_pP^{-1}C_p^\top P^\top + \frac{P}{2}\right\|}. \tag{3.35}
$$

Furthermore, given the choice made for $u(\bar{x}(t))$ in (3.26), the closed-loop vehicular system in (3.6) is also globally asymptotically stable according to hypothesis (A.13) of Theorem 7.

(*Necessity*). System (3.6) is asymptotically stable for any time delay $\tau_p(t) < \tau^\star, p = 1, ..., m$. Letting $\tau_p(t) = 0$ $(p = 1, ..., m)$ from (3.17) follows that the system $\dot{\bar{x}}(t) = F\bar{x}(t)$ with F Hurwitz, is asymptotically stable. Since F is Hurwitz, matrix $\widehat{K}$ in (3.19) is Hurwitz. Hence, applying Lemma 1, the theorem is proven. □

Note that the bound on the delay can be easily computed according to (3.35) where matrices S_p and Q act as additional degrees of freedom for the control designers to select the attainable delay bound necessary to analytically guarantee global asymptotic stability. Krasovskii methods are in the general less conservative than others with respect to the delay bound estimation [47].

3.2.3 Exponential stability

In what fallows we prove that the desired behavior can be reached not only asymptotically, but also with an exponential convergence rate that can be also estimated and calibrated with a proper gains selection. This result provides additional information on the platoon dynamics during formation and transient maneuvers. The idea is to tune the control gains within the consensus region to accomplish further requirements on platoon performance during transients under the hypothesis of the following theorem.

Theorem 2. *Consider the delayed closed-loop system in (3.6) and take the control parameters in* (3.3) *as in Lemmas 1 and 2 so that consensus is guaranteed according to Theorem 1. Assume delays $\tau_p(t)$ $(p = 1, ..., N)$ to be bounded. If there exists a constant $\delta > 0$, $\psi > 0$ and matrices $P = P^\top > 0$, $Q_p > 0$, $R_p > 0$ $(p = 1, .., m)$ so that the following inequality holds:*

$$\left[2\delta P + PA_0^\top + PA_0 + \sum_{p=1}^{m} Q_p + \psi \sum_{p=1}^{m} R_p \right] < 0, \tag{3.36}$$

being the matrix A_0 as in (3.7)*, then the closed-loop delayed system (3.6) is exponentially stable (see Definition 4).*

Proof. To show the exponential stability we consider the following Lyapunov-Krasovskii functional:

$$V(x_t) = V_1(x(t)) + V_2(x_t) + V_3(x_t), \tag{3.37}$$

where $x_t = x(t+\theta), \theta \in [-\tau^\star, 0]$ and

$$\begin{aligned}
V_1(x(t)) &= \left(x(t)^\top P x(t)\right), \\
V_2(x_t) &= \left(\sum_{p=1}^{m} \int_{t-\tau_p(t)}^{t} e^{2\delta(\theta-t)} x(\theta)^\top Q_p x(\theta) d\theta\right), \\
V_3(x_t) &= \sum_{p=1}^{m} \left(\frac{1}{1-\mu} \int_{t-\tau_p(t)}^{t} e^{2\delta(\theta-t)} x(\theta)^\top R_p x(\theta) d\theta\right).
\end{aligned} \tag{3.38}$$

Following the approach in [58], it is easy to verify that functional in (3.38) satisfies condition (A.11) of Theorem 7 as

$$\alpha_1 \|x(t)\|^2 \le V(x_t) \le \alpha_2 \|x_t\|^2 \tag{3.39}$$

being α_1, α_2 the following positive constants:

$$\begin{aligned}
\alpha_1 &= \lambda_{min}(P) \\
\alpha_2 &= \lambda_{max}(P) + \tau^\star \left[\sum_{p=1}^{m} \lambda_{max}(Q_p) + \psi \sum_{p=1}^{m} \lambda_{max}(R_p)\right]
\end{aligned} \tag{3.40}$$

with $\psi = (1-\mu)^{-1}$ being $\mu = \max_p(d_p)$ $(p = 1, \ldots, m)$. Compute now the time derivative of the terms $V_1(x(t))$, $V_2(x_t)$ and $V_3(x_t)$ in the functional (3.38). Specifically the derivative respect to time of $V_1(x(t))$ in (3.38) is:

$$\dot{V}_1(x(t)) = \dot{x}(t)^\top P x(t) + x(t)^\top P \dot{x}(t). \tag{3.41}$$

Evaluating it along the solutions of system (3.6), after some algebraic manipulation we obtain:

$$\dot{V}_1(x(t)) = x^\top(t)[A_0^\top P + PA_0]x(t) + 2x(t)^\top P \sum_{p=1}^{m} A_p x(t-\tau_p(t)). \tag{3.42}$$

The derivative respect to time of $V_2(x_t)$ is:

$$\begin{aligned}
&\dot{V}_2(x_t) = -2\delta V_2(x_t) + x(t)^\top \sum_{p=1}^{m} Q_p x(t) - \\
&-\sum_{p=1}^{m} \left((1-\dot{\tau}_p(t)) x(t-\tau_p(t))^\top Q_p x(t-\tau_p(t)) e^{-2\delta\tau_p(t)}\right)
\end{aligned} \tag{3.43}$$

and, exploiting the bound on the delay functions [39, 40], it can be recast as:

$$\begin{aligned}
&\dot{V}_2(x_t) \le -2\delta V_2(x_t) + x(t)^\top \sum_{p=1}^{m} Q_p x(t) - \\
&-\sum_{p=1}^{m} \left(\psi^{-1} x(t-\tau_p(t))^\top Q_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\right).
\end{aligned} \tag{3.44}$$

Analogously, exploiting again the bounds on the delay functions, for the derivative respect to time of $V_3(x_t)$ we have:

$$\begin{aligned}\dot{V}_3(x_t) &= -2\delta V_3(x_t) + \psi x(t)^\top \sum_{p=1}^{m} R_p x(t) + \\ &- \sum_{p=1}^{m} \Big(\psi(1-\dot{\tau}_p(t)) x(t-\tau_p(t))^\top R_p x(t-\tau_p(t)) e^{-2\delta\tau_p(t)}\Big) \\ &\leq -2\delta V_3(x_t) + \psi x(t)^\top \sum_{p=1}^{m} R_p x(t) + \\ &- \sum_{p=1}^{m} \Big(x(t-\tau_p(t))^\top R_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\Big) .\end{aligned} \tag{3.45}$$

From (3.42), (3.44), (3.45), we can write the following inequality for the time derivative of the Lyapunov-Krasovskii functional in (3.39):

$$\begin{aligned}&\dot{V}(x_t) \leq x^\top(t)[A_0^\top P + PA_0]x(t) + 2x(t)^\top P \sum_{p=1}^{m} A_p x(t-\tau_p(t)) + \\ &-2\delta V_2(x_t) + x(t)^\top \sum_{p=1}^{m} Q_p x(t) + \\ &- \sum_{p=1}^{m} \Big(\psi^{-1} x(t-\tau_p(t))^\top Q_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\Big) + \\ &-2\delta V_3(x_t) + \psi x(t)^\top \sum_{p=1}^{m} R_p x(t) - \sum_{p=1}^{m} \Big(x(t-\tau_p(t))^\top R_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\Big) .\end{aligned} \tag{3.46}$$

Now, summing and subtracting the same quantity, namely $2\delta V_1(x(t))$, after some algebraic manipulation we obtain:

$$\begin{aligned}&\dot{V}(x_t) + 2\delta V(x_t) \leq x^\top(t)[A_0^\top P + PA_0]x(t) + \\ &+2x(t)^\top P \sum_{p=1}^{m} A_p x(t-\tau_p(t)) + +2\delta x(t)^\top P x(t) + \\ &+x(t)^\top \sum_{p=1}^{m} Q_p x(t) - \sum_{p=1}^{m} \Big(\psi^{-1} x(t-\tau_p(t))^\top Q_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\Big) + \\ &+\psi x(t)^\top \sum_{p=1}^{m} R_p x(t) - \sum_{p=1}^{m} \Big(x(t-\tau_p(t))^\top R_p x(t-\tau_p(t)) e^{-2\delta\tau^\star}\Big) .\end{aligned} \tag{3.47}$$

Defining a new augmented delayed state vector as:

$$\xi(t) = [x(t), x(t-\tau_1(t)), \ldots, x(t-\tau_m(t))]^\top \tag{3.48}$$

it is possible to recast inequality (3.47) as:

$$\dot{V}(\xi) + 2\delta V(\xi) \leq \xi^\top(t) \Pi \xi(t), \tag{3.49}$$

where

$$\Pi = \begin{bmatrix} \Pi_1 & 2PA_1 & \cdots & \cdots & 2PA_m \\ 0_{3N\times 3N} & \Pi_2 & 0_{3N\times 3N} & \cdots & 0_{3N\times 3N} \\ \vdots & 0_{3N\times 3N} & \Pi_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_{3N\times 3N} \\ 0_{3N\times 3N} & \cdots & \ddots & \ddots & \Pi_{m+1} \end{bmatrix} \tag{3.50}$$

being

$$\begin{aligned}
&\Pi_1 = 2\delta P + PA_0^\top + PA_0 + \sum_{p=1}^{m} Q_p + \psi \sum_{p=1}^{m} R_p,\\
&\Pi_2 = -e^{-2\delta\tau^\star}(\psi Q_1 + R_1),\\
&\Pi_3 = -e^{-2\delta\tau^\star}(\psi Q_2 + R_2),\\
&\vdots\\
&\Pi_{m+1} = -e^{-2\delta\tau^\star}(\psi Q_m + R_m).
\end{aligned} \tag{3.51}$$

To guarantee the exponential stability, we have to show that matrix (3.50) is negative defined. This can be achieved by opportunely selecting matrices Q_p and R_p ($p = 1, ..., m$) so that the element Π_1 in (3.50) is negative definite, i.e.:

$$\Pi_1 = \left[2\delta P + PA_0^\top + PA_0 + \sum_{p=1}^{m} Q_p + \psi \sum_{p=1}^{m} R_p\right] < 0. \tag{3.52}$$

From (3.49) we now have:

$$\dot{V}(x_t) + 2\delta V(x_t) \leq 0 \tag{3.53}$$

and hence

$$V(x_t) \leq V(\phi)e^{-2\delta t} \text{ for } t \geq 0. \tag{3.54}$$

From (3.39) and (3.54) we have [58]

$$\alpha_1 \|x(t;\phi)\|^2 \leq V(x_t) \leq V(\phi)e^{-2\delta t} \leq \alpha_2 e^{-2\delta t}\|\phi\|_c^2, \tag{3.55}$$

being ϕ the functional initial conditions (see Section for nomenclature and definitions). Hence

$$\|x(t;\phi)\| \leq \sqrt{\frac{\alpha_2}{\alpha_1}} e^{-\delta t}\|\phi\|_c \text{ for } t \geq 0, \tag{3.56}$$

which conclude the theorem proof. □

Note that the decreasing exponential can be opportunely shaped by properly setting parameters δ, α_1 and α_2. Results in Section 3.2.5 show how this choice can be made so to avoid braking-critical and warning-critical distances during transient manoeuvres.

3.2.4 Disturbance Propagation Through The String

A common robustness issue in platooning is to attenuate spacing errors downstream the traffic flow [110]. According to this issue a tuning criteria for the control gains within the stability region is provided here. Following the approach in [106], [91] the analysis is carried out in the Laplace domain setting all delays to their maximum admissible value $\tau^\star$ and refers to an homogeneous traffic with a leader-predecessor configuration in the presence of a periodic perturbation acting on the leader motion. The string stability is defined here w.r.t. the vehicles positions [91] as

$$\left\| \frac{E_i(s)}{E_{i-1}(s)} \right\|_\infty \leq 1, \tag{3.57}$$

where E_i is the spacing error of vehicle i with respect to its preceding $(i = 1, ..., N)$.

Writing the vehicles dynamics (3.1) and the distributed coupling protocol (3.3) in the Laplace and approximating the constant time delay $\tau^\star$ by using a first-order Padé approximation, after algebraic manipulations, the spacing errors can be computed in terms of sensitivity functions $T_i(s)$ and $S_i(s)$ $(i = 2, .., N)$ as

$$E_i(s) = T_i(s)E_{i-1}(s) + S_i(s)\frac{d^{st}_{i,i-1}}{s}. \tag{3.58}$$

The propagating errors along the string are then attenuated by selecting the control law gains in the parameter regions that guarantees closed-loop stability, so that $|T_i(j\omega)| < 1$ for all frequencies of interest [91]. The mathematical expressions for $T_i(s)$ and $S_i(s)$ and their analytical derivation is reported in the Appendix B.

3.2.5 Numerical Results

Network and Vehicular scenario

To investigate the effectiveness of the approach we use Plexe, a platooning simulator developed in [103] (`http://plexe.car2x.org`) that updates the Veins framework. Plexe permits the analysis of platooning (i.e., automated car-following) systems performance during cooperative manoeuvres by coupling realistic vehicle dynamics (which include realistic traction force and actuation lag, i.e. the delay between the control decision and its actual realization in the vehicle due to inertial and mechanical limits) with realistic wireless network simulation. In the analysis the exemplar platoon is composed by 7 vehicles plus a leader. The exemplar topology is the one considered in [32] (see Figure 3.1), but we remark that the algorithm convergence is not restricted to this specific case and that platoon stability is guaranteed for all those topologies that satisfy hypotheses of Theorem 1.

Table 3.1: Simulation parameters for the analysis

Parameter	Value
Freeway length	10[km]
Lanes	4 (two-way)
Cars' consensus speed	100 [km/h]
Platoon size	8 cars
Platooning car max acceleration	2.5 [m/s^2]
Platooning car drivetrain constant T_i [68, 100]	$T_0 = 0.5$, $T_1 = 0.5$, $T_2 = 0.4$ $T_3 = 0.3$, $T_4 = 0.4$, $T_5 = 0.35$ $T_6 = 0.5$, $T_7 = 0.3$
Platooning car length l_i	$4 \leq l_i \leq 5$ [m]
Control Parameters in Convergence Analysis	
Headway time h_{ij}	0.8 [s]
Control gains k_{ij}	$k_{10} = 0.5$, $k_{i0} = 0.5\,(i \neq 0, i \neq 1)$ $k_{i,i-1} = 0.5$, $k_{ij} = 0$ otherwise
Control gains b_i	$b_i = 1.75$
Control gains γ_i	$\gamma_i = 0.75$
Distance at standstill d^{st}	15 [m]
Gilbert-Elliott channel	
PER p (GOOD)	0.4
PER p (BAD)	0.7
state duration	$\sim \exp(0.5s^{-1})$ ($\mathbb{E}[X] = 2s$)
CI - Non-Dimensional Warning Index	
η_{norm}	0.9
η_{min}	0.2
a_{max}	8 [m]/[s^2]
$T_{s,delay}$	0.5 [s]
$T_{h,delay}$	0.5 [s]

Vehicles and control parameters values are reported in Table 3.1. Control parameters have selected according to the theoretical results in Theorems 1 and 2 and section 3.2.4, so to induce consensus as well as good performance during transients and in the presence of periodic perturbations. The theoretical upper bound $\tau^{\star} = 20 \cdot 10^{-2}$ [s] has been computed as in (3.35). Note that the delay bound predicted trough the Krasovskii approach is above the average end-to-end communication delay typical of IEEE802.11p vehicular networks (which is of the order of hundredths of a second i.e., 10^{-2} [s] [5]) and so guarantees a certain margin of stability robustness.

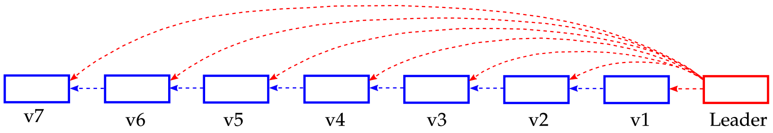


Figure 3.1: Vehicular topology in the simulation scenario.

3.2.6 Results

To show the effectiveness and robustness of the proposed approach, in this section we analyze the dynamic behavior of the platoon for different cooperative driving manoeuvres exploiting the functionalities provided by the vehicular simulator Plexe. The results presented in what follows have been obtained supposing to employ the model in presented in (2.5).

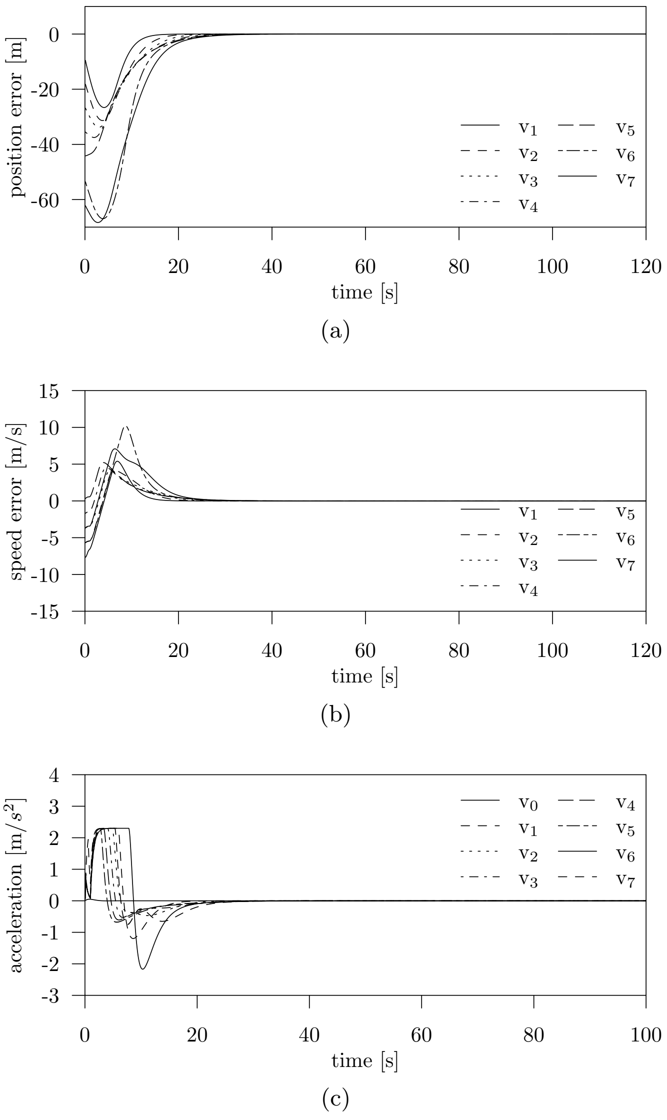
Platoon creation and maintaining

The platoon has to reach and than maintain the reference behaviour imposed by the leader moving with a constant velocity and it has to fulfil, at the same time, the constraints imposed by the desired spacing policy. Results in Figure 3.2 show that according to the theoretical derivation, vehicles starting form different initial conditions reach *consensus* and maintain the requested mutual behavior.

Consensus in the presence of Packet Losses

Although the theoretical derivation guarantee the achievement of consensus in the presence of time-varying communication delays, further results analyzes the performance of the proposed control approach in an even more realistic scenario that considers Packet Losses. Hence, in the network scenario we consider the correlated losses by using a Gilbert-Elliott channel driven by a two-state Markov chain. Each state represents the current channel status, which can be either in good of or in bad conditions (from 30% to 70% of packets lost). The channel conditions determine the Packet Error Rate Packet Error Rate (PER) to be used, enabling the possibility to simulate burst errors. State durations are drawn from an exponential distribution. Network parameters are in Table 3.1.

Figures 3.3a, 3.3b, 3.3c show position and speed errors as function of time, proving that consensus can be reached in this setup as well. Nevertheless, there are minor imperfections caused by packet losses, but position error is however in the order of centimeters, thus the system can still be considered safe and robust.



(a)

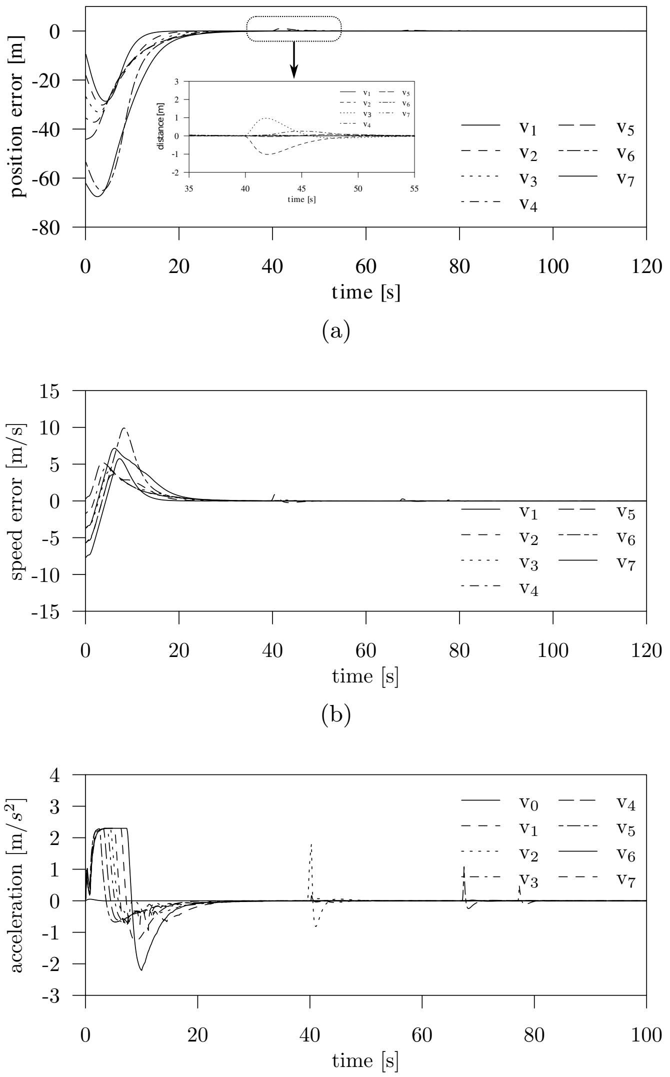
(b)

(c)

Figure 3.2: Platoon creation and maintenance. (a) Time history of the position errors $(r_i(t) - r_0(t) - h_{i0}v_i - d_{i0}^{st})$. (b) Time history of the speed errors with respect to the leader $(v_i(t) - v_0)$. (c) Time history of the vehicles accelerations $(a_i(t) - a_0)$

Leader tracking and braking maneuvers

During transient manoeuvres (e.g. decelerations due to sudden traffic jam) followers have to correctly track the time-varying leader velocity, $v_0(t)$. Results in Figure 3.4 illustrate (through the exemplar case of a leading vehicle accelerating and decelerating according to a trapezoidal speed profile)



(a)

(b)

(c)

Figure 3.3: Consensus in presence of packet losses. Gilbert-Elliott transmission channel. (a) Time history of the position errors $(r_i(t) - r_0(t) - h_{i0}v_i - d_{i0}^{st})$. (b) Time history of the speed errors with respect to the leader $(v_i(t) - v_0)$. (c) Time history of the vehicles accelerations $(a_i(t) - a_0)$

the ability of the proposed strategy in *tracking the leader* motion. Starting from the consensus condition (the platoon is engaged), the leading vehicle begins to accelerate at 60 [s] reaching a final speed of 42 [m/s]. From this condition, at 90 [s], the leader brakes keeping a constant acceleration of 1.5 [m/s^2] until it (and hence the platoon) rests (e.g. platoon at standstill).

All vehicles correctly follows during the whole manoeuvre, meeting at the same time the spacing policy constraints and guaranteeing the standstill distances when they rest. Note that, according to the theoretical derivation, control gains has been tuned inside the consensus region so to guarantee the exponential stability of the closed-loop delayed system and to avoid braking-critical and the warning-critical distances during transients.

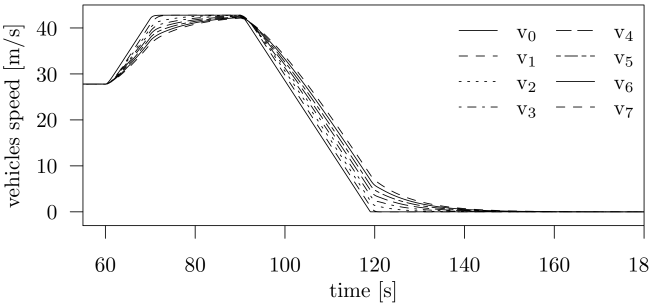


Figure 3.4: Leader tracking maneuver. Time history of the vehicles speed.

To further analyse the possible emergence of critical driving situations during transient manoeuvres, in what follows we exploit a non-dimensional warning index (or collision index CI) used in the automotive literature [86] as:

$$CI = \frac{c - d_{br}}{d_w - d_{br}} \tag{3.59}$$

where c is the actual vehicle spacing, while d_{br} and d_w are the braking-critical and warning-critical distances defined as:

$$\begin{aligned} d_{br} &= v_{rel}T_{s,delay} + f(\eta)\left(\frac{v_i^2 - (v_i - v_{rel})^2}{2a_{max}}\right) \\ d_w &= v_{rel}T_{s,delay} + f(\eta)\left(\frac{v_i^2 - (v_i - v_{rel})^2}{2a_{max}}\right) + v_iT_{h,delay} \end{aligned} \tag{3.60}$$

being v_i is the speed of i-th vehicle; $v_{rel} = v_i - v_{i-1}$ is the relative velocity between the (i)-th vehicle with respect to the $(i-1)$-th; $T_{s,delay}$ is the system delay, which is given by the brake-system hardware; a_{max} is the maximum deceleration of vehicle under normal road conditions; $T_{h,delay}$ is the delay in human response; $f(\cdot)$ is the friction scaling function and η is the estimated value of the tire-road friction coefficient. (Note that the well-known piecewise linear function depending normal friction coefficient and the smallest friction coefficients, say η_{norm} and η_{min}, is used to mimic the scaling function f [71].) All parameters can be estimated from the kinematics of couples of vehicles that brake to a full stop [86] and their values in our vehicular scenario are reported in Table 3.1.

According to (3.59) is easy to note that if c exceeds d_{br} and d_w then the warning index CI is positive, else if c is below d_{br}, then CI assume a negative value and it indicates dangerous driving situations.

In Figure 3.5 the values assumed by the warning index CI for each pair of neighbours vehicles during the trapezoidal speed profile manoeuvre in 3.4 are reported. Each box plot show that the mean value of the index is always positive (with a good safety margin) during the driving and hence the analysis confirms the safe platoon behaviour during acceleration and braking manoeuvres.

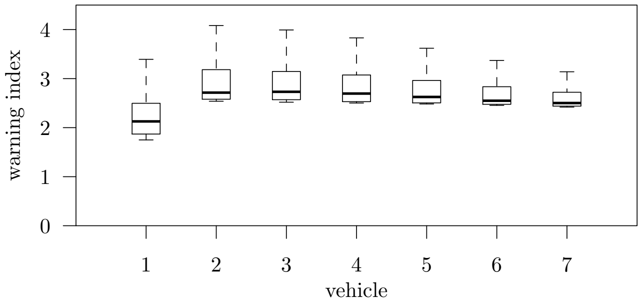
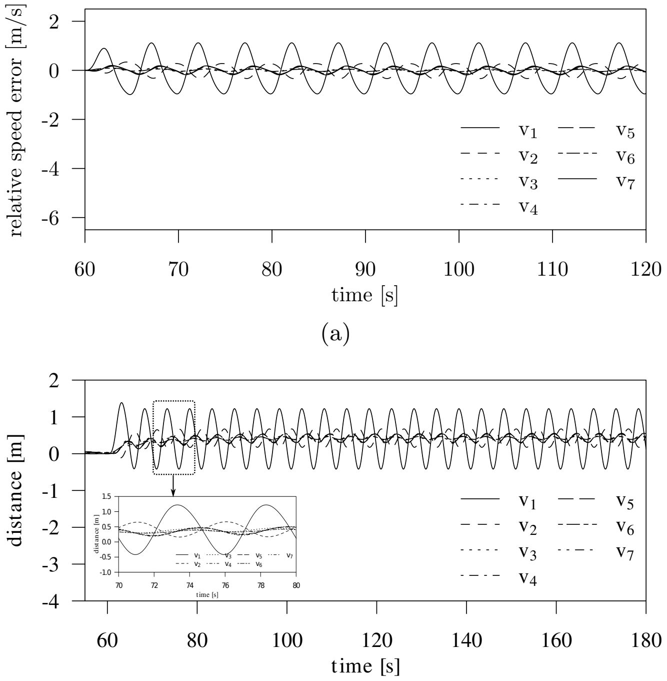


Figure 3.5: Box plot of CI for the leader tracking maneuver in 3.4.

Leader Tip-in/Tip-out

To mimic the typical leader tip-in/tip-out a periodic disturbance is added on the leader motion. Accelerate/decelerate manoeuvres obviously may generate drivetrain torques transients which may produce an objectionable disturbance to vehicle occupants or even induce instability at the ensemble level. To avoid these unwanted critical phenomena, velocity and acceleration errors have to be not amplified toward the platoon tail (string stability). Results in Figure 3.6 show that the control approach is able to counteract the effects of the periodic disturbance and, hence, errors are not amplified downstream the traffic flow. (Similar results can be obtained for the other state variable, but are omitted here for the sake of brevity.)

Further results refers to the case of a sinusoidal disturb acting on the leader acceleration due for example to aero-dynamical effects. Also in this case the proposed control approach guarantees the achievement of string stable behaviour as shown in Figure 3.2.6. (Similar behavors can be obtained for the other state variables, but again they are omitted here for the sake of



(a)

(b)

Figure 3.6: Sinusoidal disturbance acting on the leader speed, $d_a(t) = 2.7\cos(0.4\pi\mathrm{t})$, $[m/s]$. (a) Time history of vehicles speed error. (b) Time history of bumper to bumper distance computed as $r_{i-1}(t) - r_i(t) - l_{i-1}$.

brevity.)

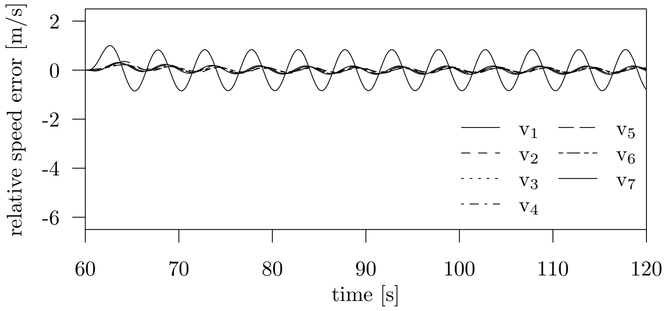
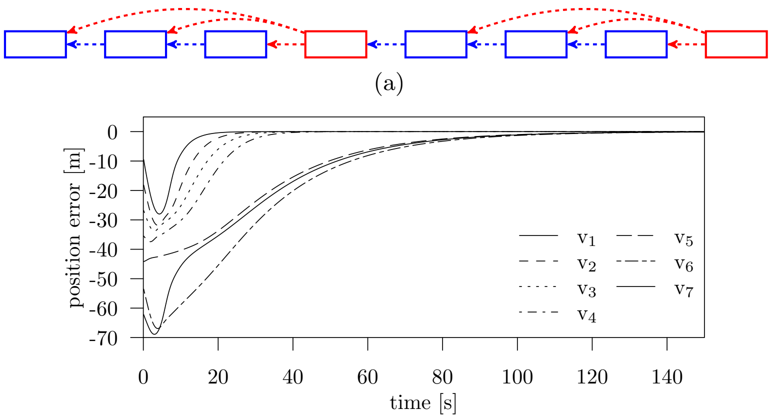


Figure 3.7: Sinusoidal disturb on the leader acceleration, $d_a(t) = 1.5\sin(0.4\pi\mathrm{t})$, $[m/s^2]$. Time history of vehicles speed error.

Joining maneuver of two mini platoons

As already mentioned, classical Cooperative Adaptive Cruise Control algorithms are usually designed –and then tuned– according to a given communication architecture, as, for example, predecessor-following or symmetric bi-directional. Since our consensus-based approach funds on inter-vehicular-communication (IVC), it allows an enhancement with respect to this specific aspect, since the algorithm deals with network topologies that are not fixed a priori and can be reconfigured depending on the actual network status without requiring modifications in the control structure, or a new gains selection. As an example, here we consider the joining maneuver of two mini platoons where vehicles communicate in clusters depending on the communication range (see Fig. 3.8a). Both mini platoons reach consensus and converge toward the desired positions and velocity without any change in the control gains values that are again set as in Table 3.1 (see results in Fig. 3.8b for the time-history of positions errors. Similar performance are achieved for the speed error).



(a)

(b)

Figure 3.8: Mini platoons. (a): Control topology. (b): Consensus: position errors computed as $r_i(t) - r_0(t) - h_{i0}v_i - d_{i0}^{st}$.

3.3 Platooning as a Synchronization Problem

In literature different approach have been used to solve synchronization problem for delayed network. One of these is the adaptive approach, used to make the complex networks robust to uncertain [126,131]. However, only few works consider the presence of time varying heterogeneous delay and only in some case the consider high order dynamics in the presence of time varying delay [135, 135]. Synchronization strategy becomes important in manoeuvres as join, split and deceleration to change lane and improving safety [74]. This kind of control allows the platoon to track the desired velocity within a given error bound. Whenever safety is not compromised, the controller will attempt to achieve the target velocity and separation from the platoon ahead in minimum time and by using acceleration and jerk within comfort limits. The cost of improved safety and comfort is in the increased time that a manoeuvre takes to be completed [6].

In what follow we will show the stability of a group of N vehicles plus a leader moving straightforward along an highway lane under the action of an adaptive control law. Indeed the presence of adaptive gains ensures the robustness of systems to different uncertainties, as not modelling dynamics and uncertain on time-varying delay. It is well-known that time-delays and uncertainties are frequently encountered in various engineering objects; the existence of time-delays and uncertainties often leads to the degradation of performance and/or instability of systems [54]. Among the different approaches to ensure robustness, the adaptivity is one of the most used [54, 126, 131].

The longitudinal dynamics of a vehicle platoon is mainly governed by the drivetrain dynamics of each vehicle (see Eq. (3.1)) and the inter-vehicle communication is modelled by a graph where every vehicle is a node (see Sec. 3.1.1). Moreover we assume that the leader dynamics is described by the following differential equations:

$$\begin{aligned} \dot{r}_0 &= v_0(t); \\ \dot{v}_0 &= a_0(t); \\ \dot{a}_0 &= -\frac{1}{T_0} a_0(t) \end{aligned} \tag{3.61}$$

The platooning control goal can be expressed as the following third-order network synchronization problem:

$$\begin{aligned} &\lim_{t\to\infty} \|r_i(t) - r_0(t) - d_{i0}\|_2 = 0, \\ &\lim_{t\to\infty} \|v_i(t) - v_0(t)\|_2 = 0, \qquad\qquad i = 1, 2, ..., N; \\ &\lim_{t\to\infty} \|a_i(t) - a_0(t)\|_2 = 0, \end{aligned} \tag{3.62}$$

where d_{i0} is the desired distance of vehicle i from the leader. The synchronization goal Eq. (3.62) is achieved here by using an appropriate distributed strategy, depending from local state variables as well as from the information received by the neighbouring vehicles (e.g., within the transmission range). The adaptive protocol proposed is the following:

$$\begin{aligned} u_i(t) = & -b_i(t)\left(v_i(t) - v_0(t)\right) - \gamma_i(t)\left(a_i(t) - a_0(t)\right) - \\ & -\frac{1}{\Delta_i}\sum_{j=0}^{N}\alpha_{ij}k_{ij}(t)(r_i(t) - r_j(t-\tau_{ij}(t)) - d_{ij}) + \\ & +\frac{1}{\Delta_i}\sum_{j=0}^{N}\alpha_{ij}k_{ij}(t)\tau_{ij}(t)v_0(t) \end{aligned} \tag{3.63}$$

where $k_{ij}(t)$, $b_i(t)$ and $\gamma_i(t)$ are the estimate of $k^{\star}_{ij} > 0$, $b^{\star}_i > 0$ and $\gamma^{\star}_i > 0$; and are updated according with the following adaptive algorithm [134]:

$$\begin{cases} \dot{\gamma}_i(t) & = \zeta_i\left(\bar{a}_i(t)\right)^2 \\ \dot{b}_i(t) & = \beta_i\left(\bar{v}_i(t)\right)^2 \\ \dot{k}_{ij}(t) & = \chi_{ij}\left(\bar{r}_i(t)\right)^2 \end{cases} \tag{3.64}$$

$\forall i = 1, ..., N$, $j = 0, .., N$ with $\zeta_i, \beta_i, \chi_{ij} \in \mathcal{R}$ positive constants gains and initial condition $k_{ij}(0) > 0$, $b_i(0) > 0$, $\gamma_i(0) > 0$.

Furthermore α_{ij} models the network topology emerging from the presence/absence of a communication link between vehicles i and j (we assume $\alpha_{0j} = 0$, $\forall j = 0, ..., N$, since the leader does not receive data from any other vehicle); $\Delta_i = \sum_{j=0}^{N}\alpha_{ij}$ is the degree of vehicle/agent i; $\tau_{ij}(t)$ represent the communication time-varying delay affecting the communication with the i-th vehicle when information is transmitted from vehicle j-th (in general $\tau_{ij}(t) \neq \tau_{ji}(t)$); d_{ij}, finally, is the desired distance between vehicles i and j according to the selected spacing policy [100].

All agents in the networks share their state information (e.g., the absolute position, the velocity and the acceleration) with all other agent through a V2V communication paradigm [25]. The on-board integration of inertial sensors with a GPS receiver allows every agent to measure its absolute position. This information is transmitted via a BSM to its nearby vehicles. On receiving each others GPS positions and relative positions, each equipped agent estimates the current positions of its own nearby vehicles [41]. Note that in the control protocol Eq. (3.63) we take into account not only the information about agent's position, but also the information about leader speed and acceleration. Both this latter information are computed on line by each agent from the information of leader position as can bee seen in [22,36,74]. From vision data, speed and acceleration are computed by decomposing the homograph [12].

The presence of time-varying gains ensures the robustness of systems to different uncertainties, as unmodeled dynamics and uncertain on time-varying delay. It is well-known that time-delays and uncertainties are frequently encountered in various engineering objects; the existence of time-delays and uncertainties often leads to the degradation of performance and/or instability of systems [55]. Among the different approaches to ensure robustness, the adaptivity is one of the most used [55, 126, 131].

Moreover, the delay $\tau_{ij}(t)$ can be assumed to be bounded as $0 \leq \tau_{ij}(t) \leq \tau^{\star}$ [15]. Note that $\tau_{ij}(t)$ is detectable and it can be evaluated for a link when the information is received, since each vehicle transmits a GPS timestamp $\bar{t}$ (i.e., the time instant when the information is sent) [23]. Note that relative information with respect to the predecessor can be also collected by on-board sensors (like radar, camera), implementing the convergence of sensor-based technologies and connected-vehicle communications for the information gathering. In this case, the delay affecting measurements collected by on-board sensors (like, e.g., radars) is negligible.

3.3.1 Delayed Vehicular Network

The time behaviour of each vehicle of the platoon can then be described by the dynamics of distance, velocity and acceleration errors. For each vehicle the error is defined as the difference between the actual distance to the leader and a fixed reference distance.

To this aim, we define position, velocity and acceleration errors with respect to the reference signals $r_0(t), v_0(t), a_0(t)$ $(i = 1, \ldots, N)$ as:

$$\begin{aligned} \bar{r}_i(t) &= r_i(t) - r_0(t) - d_{i0} \\ \bar{v}_i(t) &= \dot{\bar{r}}_i(t) = \dot{r}_i(t) - \dot{r}_0(t) \\ \bar{a}_i(t) &= \ddot{\bar{r}}_i(t) = \ddot{r}_i(t) - \ddot{r}_0(t). \end{aligned} \tag{3.65}$$

The errors dynamics for each vehicle can be written as follow:

$$\begin{aligned} \dot{\bar{r}}_i(t) = &\quad \bar{v}_i(t) \\ \dot{\bar{v}}_i(t) = &\quad \bar{a}_i(t) \\ \dot{\bar{a}}_i(t) = &\quad \dot{a}_i(t) - \dot{a}_0(t) \end{aligned} \tag{3.66}$$

Now, substituting Eq. (3.1) into Eq. (3.66) we have:

$$\begin{aligned} &\dot{\bar{r}}_i(t) = \bar{v}_i(t) \\ &\dot{\bar{v}}_i(t) = \bar{a}_i(t) \\ &\dot{\bar{a}}_i(t) = -\frac{1}{T_i} a_i(t) + \frac{1}{T_i} u_i(t) - \dot{a}_0(t). \end{aligned} \tag{3.67}$$

In what follows we will consider an homogeneous platoon; this implies that all vehicles dynamics have been described by the same drivetrain constant, i.e. $T = T_i \; \forall i = 0, 1, ..., N$. However we remark that is possible to extend our approach to an heterogeneous platoon.

Taking into account the leader dynamics in (3.61), and the control input (3.63), the closed loop dynamical system (3.67) can be written as:

$$\begin{aligned} \dot{\bar{r}}_i(t) = & \; \bar{v}_i(t) \\ \dot{\bar{v}}_i(t) = & \; \bar{a}_i(t) \\ \dot{\bar{a}}_i(t) = & \; \frac{1}{T}(-a_i(t) - b_i(t)\bar{v}_i(t) - \gamma_i(t)\bar{a}_i(t) + \\ & -\frac{1}{\Delta_i}\sum_{j=1}^{N} \alpha_{ij}k_{ij}(t)(r_i(t) - r_j(t - \tau_{ij}(t)) - d_{ij})) + \frac{1}{T}a_0(t) \end{aligned} \tag{3.68}$$

The distance d_{ij} from the i-th vehicle to the j-th can be rewritten as $d_{ij} = d_{i0} - d_{j0}$ [32]. With this position, adding and substituting $r_0(t)$, we can recast the system in Eq. (3.68) as:

$$\begin{aligned} \dot{\bar{r}}_i(t) = & \; \bar{v}_i(t) \\ \dot{\bar{v}}_i(t) = & \; \bar{a}_i(t) \\ \dot{\bar{a}}_i(t) = & \; \frac{1}{T}(-b_i(t)\bar{v}_i(t) - (1 + \gamma_i(t))\bar{a}_i(t) + \\ & -\frac{1}{\Delta_i}\sum_{j=1}^{N} \alpha_{ij}k_{ij}(t)(r_i(t) - r_j(t - \tau_{ij}(t)) + \\ & -d_{i0} + d_{j0} + r_0(t) - r_0(t))) \end{aligned} \tag{3.69}$$

Since $\bar{r}_i(t) = r_i(t) - r_0(t) - d_{i0}$ and $\bar{r}_j(t) = r_j(t) - r_0(t) - d_{j0}$, we can write:

$$\begin{aligned} \dot{\bar{r}}_i(t) = & \; \bar{v}_i(t) \\ \dot{\bar{v}}_i(t) = & \; \bar{a}_i(t) \\ \dot{\bar{a}}_i(t) = & \; \frac{1}{T}(-b_i(t)\bar{v}_i(t) - (1 + \gamma_i(t))\bar{a}_i(t) + \\ & -\frac{1}{\Delta_i}\sum_{j=1}^{N} \alpha_{ij}k_{ij}(t)(\bar{r}_i(t) - \bar{r}_j(t - \tau_{ij}(t))) \end{aligned} \tag{3.70}$$

To describe the platoon dynamics in presence of the time-varying delays associated to the different links in a more compact form we define the position, speed and acceleration error vectors as $\bar{r}(t) = [\bar{r}_1, \ldots, \bar{r}_i \ldots, \bar{r}_N]^\top$, $\bar{v}(t) = [\bar{v}_1, \ldots, \bar{v}_i \ldots, \bar{v}_N]^\top$, $\bar{a}(t) = [\bar{a}_1, \ldots, \bar{a}_i \ldots, \bar{a}_N]^\top$, and the error state vector as $\bar{x}\,(t) = \begin{bmatrix} \bar{r}^\top\,(t) & \bar{v}^\top\,(t) & \bar{a}^\top\,(t) \end{bmatrix}^\top$. Moreover delays $\tau_{ij}(t)$ in Eq. (3.69) can be recast as $\tau_p(t) \in \{\tau_{ij}(t) : i, j = 1, 2, ..., N, i \neq j)\}$ for $p = 1, 2, ..., m$ with $m \leq N(N-1)$ $(0 \leq \tau_p(t) \leq \tau)$. Note that m is equal to its maximum, $N(N-1)$, if the platoon topology is represented by a directed complete graph and all time delays are different. According to the above definitions, the closed loop platoon dynamics can be represented as the following set of functional differential equations:

$$\dot{\bar{x}}(t) = A_0(t)\bar{x}(t) + \sum_{p=1}^{m} A_p(t)\bar{x}(t - \tau_p(t)), \tag{3.71}$$

where

$$A_0(t) = \begin{bmatrix} 0_{N\times N} & I_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & I_{N\times N} \\ -T\widetilde{K}(t) & -T\widetilde{B}(t) & -\widetilde{T}(t) \end{bmatrix} \tag{3.72}$$

$$A_p(t) = \begin{bmatrix} 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N} & 0_{N\times N} \\ T\widetilde{K}_p(t) & 0_{N\times N} & 0_{N\times N} \end{bmatrix} \tag{3.73}$$

with

$$T = \text{diag}\left\{\frac{1}{T}, \dots, \frac{1}{T}\right\} \in \mathbb{R}^{N\times N}; \tag{3.74}$$

$$\widetilde{T}(t) = \text{diag}\left\{\frac{(1+\gamma_1(t))}{T}, \dots, \frac{(1+\gamma_N(t))}{T}\right\} \in \mathbb{R}^{N\times N}; \tag{3.75}$$

$$\widetilde{B}(t) = \text{diag}\{b_1(t), \dots, b_N(t)\} \in \mathbb{R}^{N\times N}; \tag{3.76}$$

$$\begin{gathered} \widetilde{K}(t) = \text{diag}\left\{\tilde{k}_{11}(t), \dots, \tilde{k}_{NN}(t)\right\} \in \mathbb{R}^{N\times N} \\ \text{with } \tilde{k}_{ii}(t) = \frac{1}{\Delta_i}\sum_{j=0}^{N} k_{ij}(t)\alpha_{ij}; \end{gathered} \tag{3.77}$$

and $\widetilde{K}_p(t) = [\bar{k}_{ij,p}] \in \mathbb{R}^{N\times N}$ $(p = 1, \dots, m)$ the matrix defined according to the formalism adopted in [130] as:

$$\bar{k}_{ij,p}(t) = \begin{cases} \dfrac{\alpha_{ij}k_{ij}(t)}{\Delta_i}, & j \neq i, \tau_p(\cdot) = \tau_{ij}(\cdot), \quad i,j = 1,\dots,N \\ 0, & j \neq i, \tau_p(\cdot) \neq \tau_{ij}(\cdot), \quad i,j = 1,\dots,N \\ 0, & j = i, \quad i,j = 1,\dots,N \end{cases} \tag{3.78}$$

3.3.2 Stability Analysis

Consider the system in (3.71)

$$\dot{\bar{x}}(t) = A_0(t)\bar{x}(t) + \sum_{p=1}^{m} A_p(t)\bar{x}(t - \tau_p(t)) \tag{3.79}$$

and the following updating law for the gain (3.64) ($i = 1, ..., N$,$j = 1, ..., N$):

$$\begin{cases} \dot{\gamma}_i(t) = \zeta_i\left(\bar{a}_i(t)\right)^2 & \gamma_i(0) > 0 \\ \dot{b}_i(t) = \beta_i\left(\bar{v}_i(t)\right)^2 & b_i(0) > 0 \\ \dot{k}_{ij}(t) = \kappa_{ij}\left(\bar{r}_i(t)\right)^2 & k_{ij}(0) > 0 \end{cases} \tag{3.80}$$

Furthermore, consider the following matrix:

$$\widehat{K}(t) = \Delta\left(\tilde{K}(t) + \sum_{p=1}^{m}\tilde{K}_p(t)\right) \tag{3.81}$$

where $\Delta = \text{diag}\{\Delta_1, \dots, \Delta_N\} \in \mathbb{R}^{N\times N}$.

We now introduce one lemma that will be instrumental for the proof of convergence presented in this section.

Lemma 3. *[32] Supposing $k_i(t) = \dfrac{k_{i0}(t)\alpha_{i0}}{\Delta_i} > 0$, $\forall t \geq t_0$ and for $(i = 1, ..., N)$, the matrix $\widehat{K}(t)$ in (3.81) is positive stable if and only if node 0 is globally reachable in $\mathcal{G}_{N+1}$.*

The proof of stability is based on recasting the delayed closed-loop dynamics as a set of functional differential equations for which it is possible to find a quadratic Lyapunov-Krasovskii function [46]. Synchronization in the presence of heterogeneous time varying delay can be guaranteed under the classical constraints on bounded delay functions [40,46], i.e $\tau_p(t) \in [0, \tau_{max}]$, $\dot{\tau_p}(t) \in (0, d_p]$ $\forall t$, $\forall p$ and $d_p \leq 1$, according to the following theorem.

Theorem 3. *Consider system in Eq.* (3.71)*, under the action of the adaptive control law Eq.* (3.63)*. Assume delays $\tau_p(t)$ $(p = 1, ..., m)$ to be bounded and node 0 is globally reachable in $\mathcal{G}_{N+1}$. Then for $\tau_p(t) < \tau^\star$ $\forall p$ $\forall t$ we have that the system in (3.71) is uniformly stable.*

Proof. Define a matrix $M(t) > 0$ so that the matrix $P \in S_n^+$ is the solution of the Lyapunov equation:

$$A_0^\top(t)P + P(t)A_0(t) = -M(t) \tag{3.82}$$

according to Theorem 5. To show the stability of the system, we select a proper Krasovskii functional [134] for the system (3.71):

$$\begin{aligned} V(\bar{x}, k_{ij}, b_i, \gamma_i) = \bar{x}(t)^\top P\bar{x}(t) + \sum_{p=1}^{m}\int_{t-\tau_p(t)}^{t}\bar{x}(s)^\top R_p\bar{x}(s)ds + \\ + \sum_{i=1}^{N}\sum_{j=0}^{N}\frac{\left(k_{ij}(t) - k_{ij}^\star\right)^2}{2\kappa_{ij}} + \sum_{i=1}^{N}\frac{(b_i(t) - b_i^\star)^2}{2\beta_i} + \sum_{i=1}^{N}\frac{(\gamma_i(t) - \gamma_i^\star)^2}{2\zeta_i} \end{aligned} \tag{3.83}$$

where $R_p \in \mathbb{R}^{3N\times 3N}$, $p = 1, \dots, m$ and $P \in \mathcal{S}_n^+$ are constants symmetric and positive defined matrices to be determined. Define the following continuous, non-decreasing and positive functions according to the hypothesis of

Theorem 7 as:

$$\begin{cases} u(\|\bar{x}(t)\|) = \bar{x}(t)^\top P \bar{x}(t) \\ v(\|\bar{x}(t,\tau^\star)\|_c) = qV(\bar{x}(t,\tau^\star), k^\star_{ij}, b^\star_i, \gamma^\star_i) \end{cases} \tag{3.84}$$

being the scalar $q > 1, \in \mathbb{R}$. From definition (3.84) the condition (A.11) in Theorem 7 is satisfied, i.e. :

$$u(\|\bar{x}(\cdot)\|) \leq V(\cdot) \leq v(\|\bar{x}(\cdot)\|_c) \tag{3.85}$$

Now, differentiating Krasovskii functional in Eq. (3.83) yields:

$$\begin{aligned} \dot{V}(\cdot) &= \left(\dot{\bar{x}}(t)^\top P \bar{x}(t) + \bar{x}(t)^\top P \dot{\bar{x}}(t)\right) + \sum_{p=1}^{m} \bar{x}(s)^\top R_p \bar{x}(s)\dot{s}\Big|_{s=t-\tau_p(t)}^{s=t} + \\ &+ \sum_{i=1}^{N}\sum_{j=0}^{N} \frac{\left(k_{ij}(t) - k^\star_{ij}\right)}{\kappa_{ij}} \dot{k}_{ij}(t) + \sum_{i=1}^{N} \frac{\left(b_i(t) - b^\star_i\right)}{\beta_i} \dot{b}_i(t) + \sum_{i=1}^{N} \frac{\left(\gamma_i(t) - \gamma^\star_i\right)}{\zeta_i} \dot{\gamma}_i(t) \end{aligned} \tag{3.86}$$

Hence substituting (3.71) and (3.64) in (3.86), it follows:

$$\begin{aligned} \dot{V}(\cdot) &= \left(A_0(t)\bar{x}(t) + \sum_{p=1}^{m} A_p(t)\bar{x}(t-\tau_p(t))\right)^\top P\bar{x}(t) + \\ &+ \bar{x}(t)^\top P \left(A_0(t)\bar{x}(t) + \sum_{p=1}^{m} A_p(t)\bar{x}(t-\tau_p(t))\right) + \sum_{p=1}^{m} \bar{x}(s)^\top R_p \bar{x}(s)\dot{s}\Big|_{s=t-\tau_p(t)}^{s=t} \\ &+ \sum_{i=1}^{N}\sum_{j=0}^{N} \left(k_{ij}(t) - k^\star_{ij}\right) \bar{r}_i(t)^2 + \sum_{i=1}^{N} \left(b_i(t) - b^\star_i\right) \bar{v}_i(t)^2 + \sum_{i=1}^{N} \left(\gamma_i(t) - \gamma^\star_i\right) \bar{a}_i(t)^2. \end{aligned} \tag{3.87}$$

Introduce now the following matrices:

$$\Theta(t) = \begin{bmatrix} \widehat{K}(t) & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathrm{diag(b_i(t))} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} & \mathrm{diag(\gamma_i(t))} \end{bmatrix} > 0 \tag{3.88}$$

$$\Theta^\star = \begin{bmatrix} K^\star & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathrm{diag(b^\star_i)} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} & \mathrm{diag(\gamma^\star_i)} \end{bmatrix} > 0 \tag{3.89}$$

where $K^\star \in \mathcal{R}^{N\times N}$ is a matrix containing the positive constant $k^\star_{ij}$ $(i,j = 1,...,N)$. Note that the diagonal block matrix $\Theta(t)$ in (3.88) is positive definite for construction since $\widehat{K}(t) > 0$ according to Lemma 3 and $\mathrm{diag(b_i(t))} > 0$ and $\mathrm{diag(\gamma_i(t))} > 0$ [99]. According to definition (3.88) and (3.89), after some algebraic manipulation (3.87) can be recast as:

$$\begin{aligned} \dot{V}(\cdot) &= \bar{x}(t)^\top \left(A_0(t)^\top P + PA_0(t)\right)\bar{x}(t) + \left(\sum_{p=1}^{m} A_p(t)\bar{x}(t-\tau_p(t)\right)^\top P\bar{x}(t) + \\ &+ \bar{x}(t)^\top P \left(\sum_{p=1}^{m} A_p(t)\bar{x}(t-\tau_p(t))\right) + \sum_{p=1}^{m} \bar{x}(t)^\top R_p \bar{x}(t) + \\ &- \sum_{p=1}^{m} \left(\bar{x}(t-\tau_p(t))^\top R_p \bar{x}(t-\tau_p(t))(1-\dot{\tau}_p(t))\right) + \bar{x}(t)^\top \Theta \bar{x}(t) - \bar{x}(t)^\top \Theta^\star \bar{x}(t) \end{aligned} \tag{3.90}$$

and then as:

$$\dot{V}(\cdot) = \bar{x}(t)^\top \Bigg(A_0(t)^\top P + PA_0(t) + \Theta(t) \Bigg) \bar{x}(t) + 2\bar{x}(t)^\top \sum_{p=1}^{m} A_p(t)^\top P\bar{x}(t-\tau_p(t)) + \\ -\sum_{p=1}^{m} \Bigg(\bar{x}(t-\tau_p(t))^\top R_p \bar{x}(t-\tau_p(t))(1-\dot{\tau}_p(t)) \Bigg) + \bar{x}(t)^\top \left(\sum_{p=1}^{m} R_p - \Theta^\star \right) \bar{x}(t). \tag{3.91}$$

From (3.82) we obtain:

$$\dot{V}(\cdot) = \bar{x}(t)^\top \left(\sum_{p=1}^{m} R_p - \Theta^\star - M(t) + \Theta(t) \right) \bar{x}(t) + 2\bar{x}(t)^\top \sum_{p=1}^{m} PA_p(t)\bar{x}(t-\tau_p(t)) + \\ -\sum_{p=1}^{m} \Bigg(\bar{x}(t-\tau_p(t))^\top R_p \bar{x}(t-\tau_p(t))(1-\dot{\tau}_p(t)) \Bigg). \tag{3.92}$$

Now note that, for any generic positive definite matrix $G \in \mathcal{R}^{3N \times 3N}$, it holds [60]: $2a^\top c \leq a^\top G a + c^\top G^{-1} c$. Setting $a^\top = \bar{x}(t)^\top PA_p(t)$, $c = \bar{x}(t-\tau_p(t))$ Eq. (3.92) becomes:

$$\begin{aligned} \dot{V}(\cdot) \leq\ & \bar{x}(t)^\top \left(\sum_{p=1}^{m} R_p - \Theta^\star - M(t) + \Theta(t) \right) \bar{x}(t) + \\ & + \sum_{p=1}^{m} \Big(\bar{x}(t-\tau_p(t))^\top G \bar{x}(t-\tau_p(t)) + \bar{x}(t)^\top PA_p(t) G A_p(t)^\top P \bar{x}(t) \Big) + \\ & - \sum_{p=1}^{m} \Bigg(\bar{x}(t-\tau_p(t))^\top R_p \bar{x}(t-\tau_p(t))(1-\dot{\tau}_p(t)) \Bigg), \end{aligned} \tag{3.93}$$

and after some algebraic manipulation can be recast as :

$$\dot{V}(\cdot) \leq \bar{x}(t)^\top \Bigg(-\Theta^\star - M(t) + \Theta(t) + \sum_{p=1}^{m} R_p + \sum_{p=1}^{m} PA_p(t) G A_p(t)^\top P \Bigg) \bar{x}(t) + \\ + \sum_{p=1}^{m} \bar{x}(t-\tau_p(t))^\top \left(G - R_p(1-\dot{\tau}_p(t)) \right) \bar{x}(t-\tau_p(t)). \tag{3.94}$$

Exploiting the bound on the delay function [40, 46], it follows:

$$\dot{V}(\cdot) \leq \bar{x}(t)^\top \Bigg(-\Theta^\star - M(t) + \Theta(t) + \sum_{p=1}^{m} R_p + \sum_{p=1}^{m} PA_p(t) G A_p(t)^\top P \Bigg) \bar{x}(t) + \\ + \sum_{p=1}^{m} \bar{x}(t-\tau_p(t))^\top \left(G - R_p(1-d_p) \right) \bar{x}(t-\tau_p(t)). \tag{3.95}$$

Choosing $G = \displaystyle\sum_{p=1}^{m} R_p$. we have

$$\dot{V}(\cdot) \leq \bar{x}(t)^\top \Bigg(-\Theta^\star - M(t) + \Theta(t) + \sum_{p=1}^{m} R_p + \sum_{p=1}^{m} PA_p(t) G A_p(t)^\top P \Bigg) \bar{x}(t). \tag{3.96}$$

Selecting $\Theta^\star$ so that the matrix

$$\Lambda = -\Theta^\star - M(t) + \Theta(t) + \sum_{p=1}^{m} R_p + \sum_{p=1}^{m} PA_p(t) G A_p(t)^\top P, \tag{3.97}$$

is negative definite, we have:

$$\dot{V}(\cdot) \leq \bar{x}(t)^{\top} \Lambda \bar{x}(t) \tag{3.98}$$

Hence condition (A.12) of Theorem 7 is satisfied and the closed loop system (3.71) is uniformly asymptotically stable. In addiction for $u(s)$ defined in (3.84) it follows

$$\lim_{s \to \infty} u(s) = +\infty,$$

so the system (3.71) is globally uniformly asymptotically stable. □

3.3.3 Results

To show the effectiveness and robustness of the proposed control strategy, in this section we analyse the dynamic behaviour of the platoon for different driving manoeuvres. In the analysis an exemplar platoon composed by 7 vehicles and a leader has been considered. The chosen control topology is the predecessor-following one, considered in [90] and coherent with [100], where the leader communicates with all the vehicles in broadcast, and every vehicle shares information with its follower. We remark that the algorithm convergence is not restricted to the case of classical predecessor-following architecture based on pairwise interactions [98], but it ensures platoon stability for all those topologies that satisfy hypotheses of Theorem 3. The spacing policy requires a desired distance between vehicles i and j at standstill $d_{ij}^{st} = 15\,\mathrm{m}$. Regarding to vehicle's characteristics, we refer to an heterogeneous platoon whose drivetrain constant have been tuned according to Tab. 3.2. Simulations have been performed by the vehicular simulator Plexe. The results presented in what follows have been obtained supposing to employ the model in presented in (2.5).

Platoon creation and maintaining

In this subsection we analyse the ability of our platoon to track a constant speed reference. Starting form different initial conditions, the platoon has to reach and than maintain the reference behaviour as imposed by the leader moving with a constant velocity and has to fulfil, at the same time, the constraints imposed by the desired spacing policy. The leader vehicle imposes a common and constant fleet velocity equal to $27.7\,\mathrm{m\,s^{-1}}$ (i.e. $100\,\mathrm{km\,h^{-1}}$) and the control parameters has been chosen as in Tab. 3.2.

Table 3.2: Simulation parameters for the analysis

Parameter	Value
Freeway length	10[km]
Lanes	4 (two-way)
Platoon size	8 cars
Platooning car max acceleration	2.5 [m/s^2]
Platooning car drivetrain constant T_i $[68, 100]$	$T_0 = 0.5$, $T_1 = 0.5$, $T_2 = 0.4$ $T_3 = 0.3$, $T_4 = 0.4$, $T_5 = 0.35$ $T_6 = 0.5$, $T_7 = 0.3$
Platooning car length l_i	$4 \leq l_i \leq 5$ [m]
Control Parameters in Convergence Analysis	
Control gains k_{ij}	$\chi_{10} = 0.015$, $\chi_{i0} = 0.015\,(i \neq 0, i \neq 1)$ $\chi_{i,i-1} = 0.015$, $\chi_{ij} = 0$ otherwise
Control gains β_i	$\beta_i = 0.275$
Control gains ζ_i	$\gamma_i = 1.175$
Distance at standstill d^{st}	15 [m]

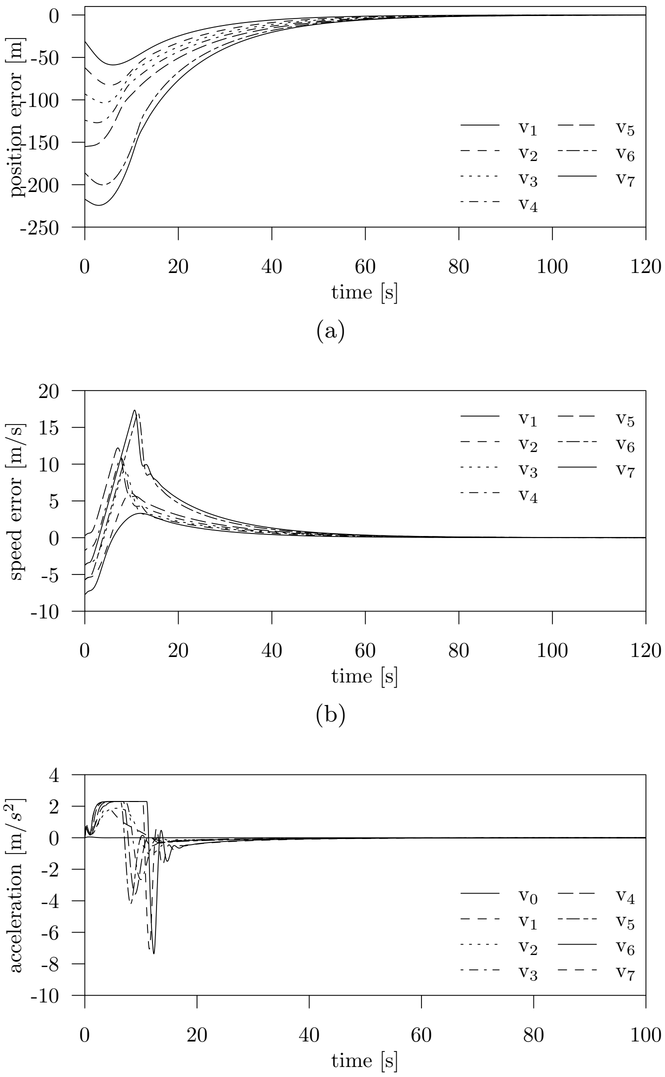
Note that the initial condition of the adaptive gains are chosen equal to zero. Results in (Fig. 3.9) show that, according to the theoretical derivation, the vehicles starting from different initial conditions, reach consensus and maintain the reference behaviour as imposed by leader, according to the desired spacing policy. The result confirm the ability of the proposed approach of creating and maintaining the platoon.

Note that also the adaptive gains (see Fig. 3.10) don't diverge but reach a constant steady-state value.

Leader Tip-in/Tip-out

To mimic the typical leader tip-in/tip-out a periodic disturbance is added on the leader motion. Accelerate/decelerate manoeuvres obviously may generate drivetrain torques transients which may produce an objectionable disturbance to vehicle occupants or even induce instability at the ensemble level. In Fig. 3.11 we show results. Starting from an engaged condition of the platoon, the proposed strategy allow followers to track the sinusoidal reference. In Fig. 3.11a we shows results obtained imposing a sinusoidal reference to the leader speed. In Fig. 3.11b we shows results obtained imposing a sinusoidal reference to the leader acceleration.

Furthermore in Fig. 3.12 we show, as exemplar case, the time history of the adaptive gain $b(t)$ imposing a sinusoidal reference to the leader speed. In this case, it's interesting to observe what happens to adaptive gains when a sinusoidal signal acts on leader motion. It is interesting to see that when



(a)

(b)

(c)

Figure 3.9: Platoon creation and maintenance: (a) time history of the position errors computed as $r_i(t) - r_0(t) - d_{i0}$; (b) time history of the speed $v_i(t)$; (c) time history of the acceleration $a_i(t)$.

the sinusoidal reference acts, the adaptive gain increase to counteract to the presence of tracking errors.

Leader tracking maneuvers

During transient manoeuvres (e.g. decelerations due to sudden traffic jam) followers have to correctly track the time-varying leader velocity, $v_0(t)$.

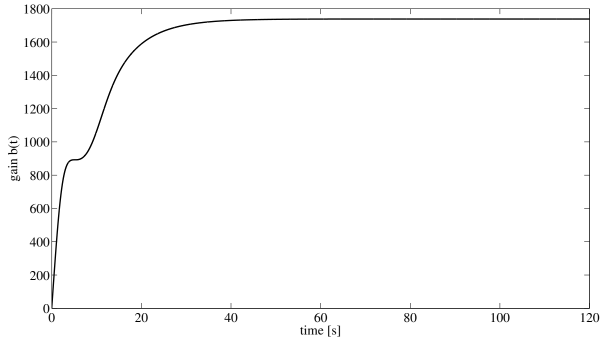
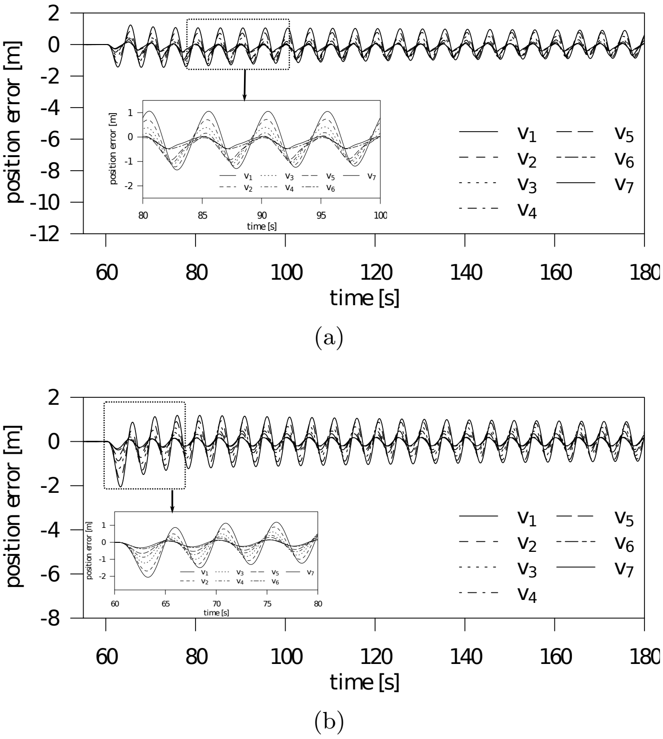


Figure 3.10: Adaptive gain: time history of $b_1(t)$.



(a)

(b)

Figure 3.11: Time history of bumper to bumper distance computed as $r_{i-1}(t) - r_i(t) - l_{i-1}$: (a) sinusoidal disturbance acting on the leader speed, $d_a(t) = 2.7\cos(0.4\pi \text{t})$, $[m/s]$; (b) sinusoidal disturb on the leader acceleration, $d_a(t) = 1.5\sin(0.4\pi \text{t})$, $[m/s^2]$.

Results in Fig. 3.13 illustrate (through the exemplar case of a leading vehicle accelerating and decelerating according to a trapezoidal speed profile) the ability of the proposed strategy in *tracking the leader* motion. Starting from

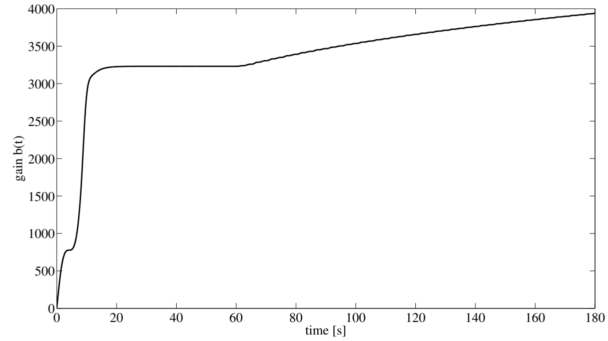
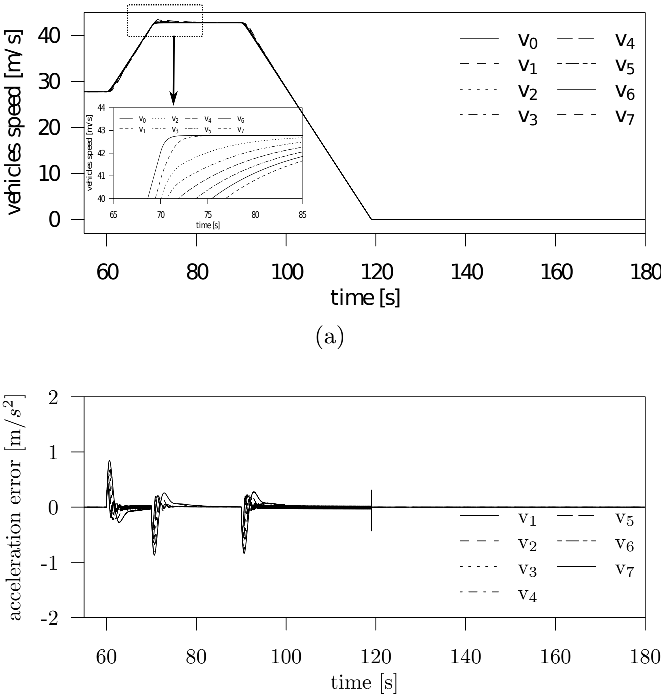


Figure 3.12: Adaptive gain: time history of $b_1(t)$.

the consensus condition (the platoon is engaged), the leading vehicle begins to accelerate at 60 [s] reaching a final speed of 42 [m/s]. From this condition, at 90 [s], the leader brakes keeping a constant acceleration of 1.5 [m/s^2] until it (and hence the platoon) rests (e.g. platoon at standstill). All vehicles correctly follows during the whole manoeuvre, meeting at the same time the spacing policy constraints and guaranteeing the standstill distances when they rest.

In Fig. 3.14 we show an exemplar case of the adaptive gain $b_1(t)$. Note that the adaptive gain, after a transient variations in correspondence of the leader tracking maneuvers, as detailed in Fig. 3.14, reach a constant steady-state value.



(a)

(b)

Figure 3.13: Leader Tracking:(a) time history of the speed $v_i(t)$; (b) time history of the acceleration error $a_0(t) - a_i(t)$.

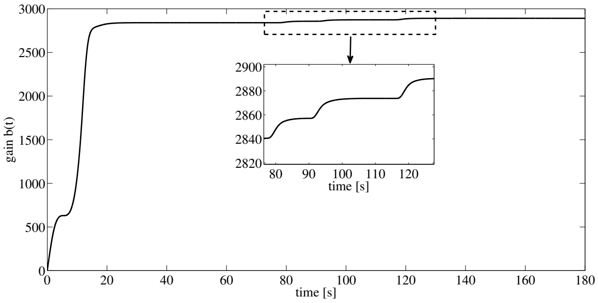


Figure 3.14: Adaptive gain: time history of $b_1(t)$.

Chapter 4

Toward the experimental validation

Abstract

Part of my PhD activities has been dedicated to company duties. One of the most challenging has been the design and implementation of an OBU, for collection and transmission of in-vehicle data (CAN data, GPS traces and so on) to neighbourhood vehicle. A desired subset of these data, opportunely encapsulated, are then forwarded to a remote system. To this purpose, a proper DSRC communication and system has been designed and implemented. To validate the effectiveness of the OBU's algorithm a proper HIL platform has been developed. The HIL platform has been designed for experimental validation of both cooperative driving and probe vehicles application, allowing user to develop control strategies and algorithm for data analysis, off-vehicle.

4.1 Introduction

Till now we have assumed standard DSRC/WAVE [2,3] access network with beaconing messages, and proper integration of different components of a cooperative driving system (emergency braking [104], anti-collision techniques, etc.), disregarding their design and implementation. However to work properly, ITS applications needs, as fundamental component, Intelligent Vehicle Technologies (IVT). IVT comprise all the electronic, electromechanical, and electromagnetic devices, operating in conjunction with on board computer controller and radio transceivers to understand the environment immediately around the vehicle, assisting the driver in operations either fully controlling the vehicle or disseminating information [13]. In this perspective, automotive vendors has started to design future vehicles (often referred as Intelligent Vehicle (IV)) equipping them with appropriate on board IVT and communication devices, necessary to create an ad-hoc wireless network connecting vehicles among them. The challenge is to create an on board communication and sensing systems capable to:

- communicate and coordinate vehicles remotely;
- provide infotainment and internet connectivity services;
- collect GPS position and in vehicle data and messages coming from the local ECUs via the Controller Area Network (CAN bus);
- forward data remotely using a data connection;
- control and operate vehicles according to desired control algorithms on the basis of local data and nearby vehicles;
- an input/output interface that allows human interaction with the local system and ITS host side applications;
- radar and camera sensors.

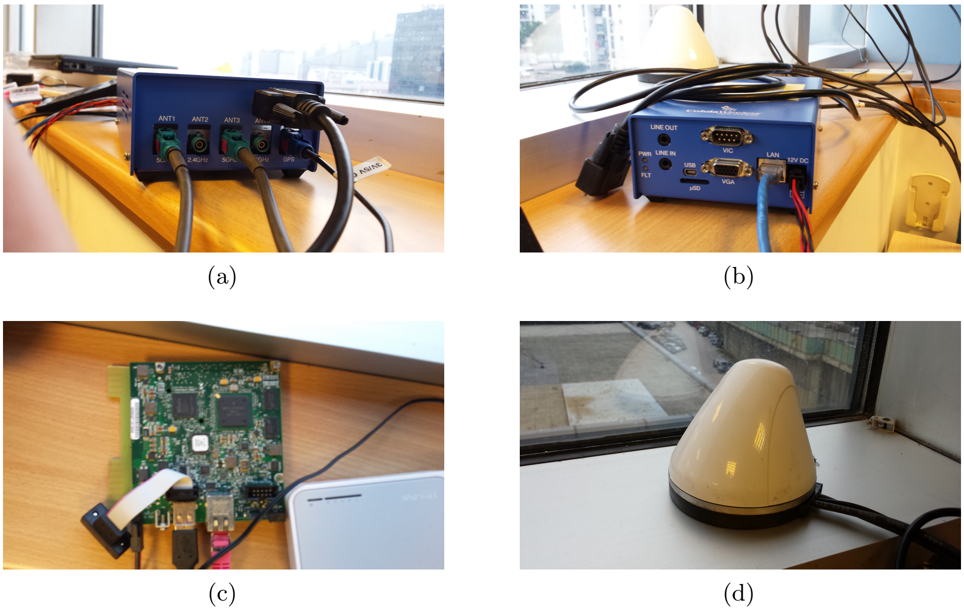
Such technological systems, integrated with in-vehicle electronic sensing systems, are often managed by a technological in silicon unit referred with the term of OBU. In what follow with the term OBU, we will refer to the technological device that allows to collect and forward data from a vehicle to a remote server (in some case referred with the term Operational Center (OC)), or to another vehicle. In the next sections we will describe the OBU and an Hardware-In-the Loop (HIL) platform, designed during my applied research activity in company. The realized OBU compute different control and data management algorithms and through an, through user-friendly interface, show a synoptic of the vehicle and the driving suggestions to the user. Furthermore, an HIL platform, has been designed to test the effectiveness of the OBU's communication and control strategy.

4.2 OBU design and implementation

The OBU, designed and realized during my research activity, has the task to monitors the real-time status of a vehicle by obtaining data and information (such as, for example, speed, battery voltage, instant fuel consumption, etc.) from the CAN bus, thanks to an OBD (On Board Diagnostic) connector. The OBU, equipped with interfaces to the bus CAN, allows users to monitor the real-time status of all the electronic control units through an user friendly interface. The goal of the OBU is to handle all the information exchanged inside and outside of vehicle, with the specific aim of collecting all these real-time information to support drivers during the driving tasks. The OBUs is an integrated system composed mainly by the following elements:

- A Cohda Wireless MK3 Cooperative-ITS Module. It includes an IEEE 802.11p radio, embedded GPS (see Fig. 4.1d) receiver and embedded processor. The embedded processor runs Linux and has USB 2.0, Ethernet, VGA and CAN interfaces;
- sbRIO National Instruments with 400 MHz processor, reconfigurable FPGA Xilinx Spartan-6 LX45, Ethernet 10/100BASE-T, CAN bus, USB and RS232 port;
- Netgear hot spot to allow wireless connectivity;
- A PDA programmed with Android technology, with Wi-Fi and 3G connectivity;
- 5 port Ethernet switch;
- A CAN cable to connect Cohda CAN interface to the in-vehicle EOBD;
- Ethernet and power cables;

In Fig. 4.2, it is possible to see a schematic representing overall OBU's architecture. sbRIO communicate with the PDA (an Android mobile device) via the wireless hotspot (Netgear) and with the Cohda MK3 via Ethernet cable. Cohda Wireless MK3, get CAN frame by the OBD diagnostic port of the vehicle through a proper designed cable. This cable, connect CAN-H, CAN-L line to the CAN transceiver of Cohda MK3, and, furthermore feeds both sbRIO and Cohda MK3 connecting them to the car's battery power supply lines. Furthermore Cohda Wireless collect GPS data, support IEEE 802.11p protocol and manage the V2V communication with other vehicles both in receiving and forwarding. The PDA provides data connectivity (mainly used to implement the V2I communication architecture), graphical user interface and forward on-board collected/computed data to an OC. sbRIO is responsible for the on-board computation of different algorithms, in



(a)

(b)

(c)

(d)

Figure 4.1: (a) Cohda Wireless: detail of the antenna and GPS plugs. (b) Cohda Wireless: detail of the CAN/Ethernet/Power supply interface. (c) sbRIO 9606: top view. (d) Cohda Wireless: GPS/radio antenna module.

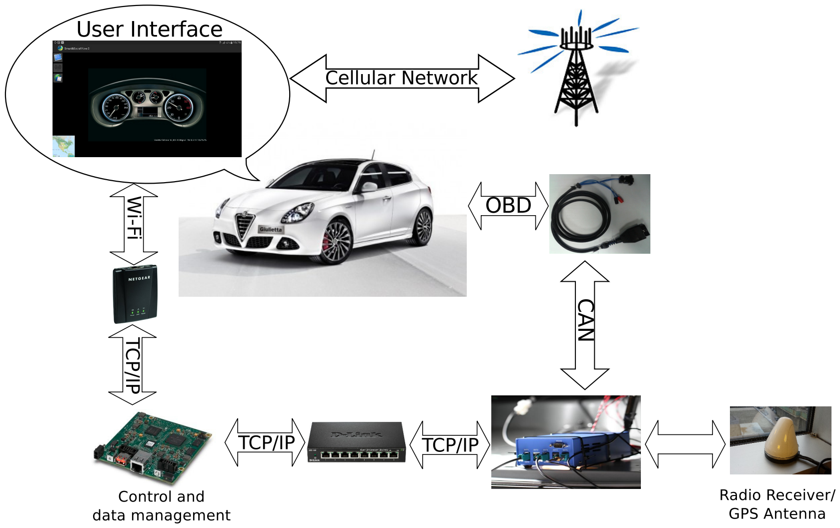


Figure 4.2: OBU's in vehicle architecture and communication paradigm.

particular: *i) Algorithm for collection of the vehicle CAN data.* Data, both GPS and CAN frame received from Cohda MK3 via the TCP/IP protocol, are collected, queued and filtered appropriately. This operation is done via software, choosing the right message (CAN frame) by address. Once a CAN frame has been filtered, this one is encapsulated, together with the GPS informations and the vehicle identification cod, into a TCP message. This message is then forwarded, via wireless connection, to the PDA. The PDA, in turn, execute two action: refresh the interactive dashboard status that reproduces the vehicle synoptic and encapsulate some specific data into a packet. This packet is then forwarded to a remote server for analysis and memorization. All of these operation are executed in real time with a period, customizable by user. *ii*) *Algorithm for fleets management.* It manages data coming from V2V communication from neighbours vehicle. The realized strategy let the opportunity to implement different local control action (e.g. standard CACC or alternative and innovative platooning strategies). Basically, the sbRIO on the basis of collected CAN frame and of Cohda's V2V and GPS data can be easily configured to compute local cooperative driving control algorithm. *iii*) *Algorithm for user-OBU interaction.* It manages the Wi-Fi connection between sbRIO and user-interface, forward data to the Android mobile device, that will use them to refresh the dashboard and to forward them in V2I communication. Cohda Wireless is responsible for communication strategies with other vehicles. An algorithm, written in C language, filter properly the CAN massage gathering together with data from GPS antenna. These data are made available via an UDP port and encapsulated into a BSM message that is forwarded in broadcast to vehicle all around. On the other hand, the radio antenna receive beacons from neighbour vehicles and put them available on an UDP port for sbRIO [1]. All of these algorithm have been tested both in HIL and experimentally (on-board of test vehicles), demonstrating the feasibility of the approach. In Fig. 4.3 is possible to see the vehicles used for the experimental activities.

4.2.1 sbRIO: detail on software implementation

As mentioned, the sbRIO handle and compute data retrieved by neighbours vehicle and data collected by on-board sensors. The application that operate these tasks has been coded in LabVIEW and can schematically divided into three main software block, each of which handles a different process:

- producer;
- consumer;

Figure 4.3: Testing in-vehicle.

- data collector.

Producer task

The producer (see Fig. 4.4) is responsible for the collection and storage of CAN data forwarded via TCP/IP by Cohda. As can be seen in Fig. 4.5, the

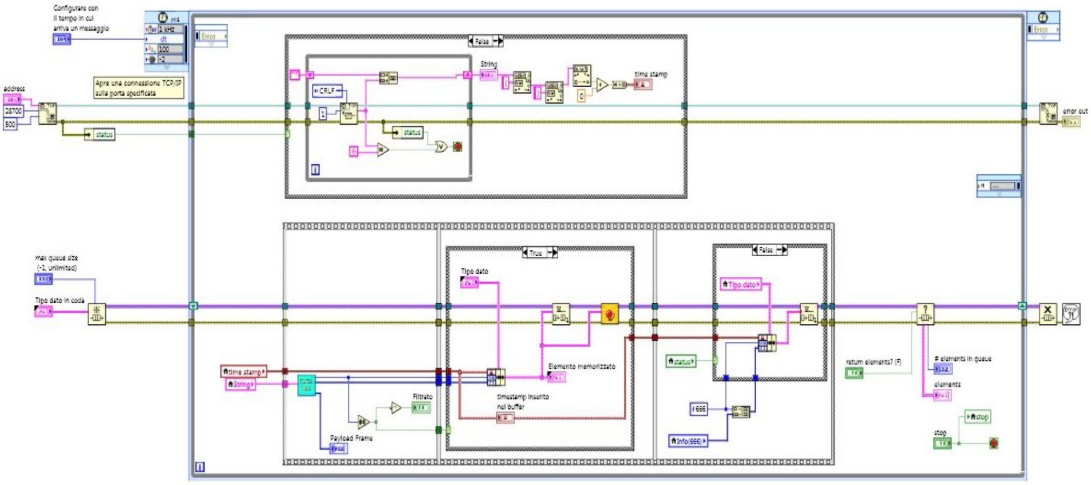


Figure 4.4: Producer: software module for CAN data collection.

first operation performed by sbRIO is to open a TCP/IP connection toward the Cohda Wireless. If the TCP/IP connection is successful established, then the producer starts the collection of data, otherwise a new connection attempt is done. If no error occurs, a cycle that read and store data (forwarded by Cohda wireless) into a queue structure will run. Data forwarded by Cohda wireless are read one byte for once. The time stamp, associated to each received packet is extracted separately (as can be seen in Fig. 4.6. Now, as depicted in Fig. 4.7, collected data are filtered and enqueued into a data structure. A software block analyse each single frame and activates an alarm flag in case of anomalies in values. Next, a message named *Info(666)* is pushed into the queue. This message contain useful data for the interaction among user and vehicle.

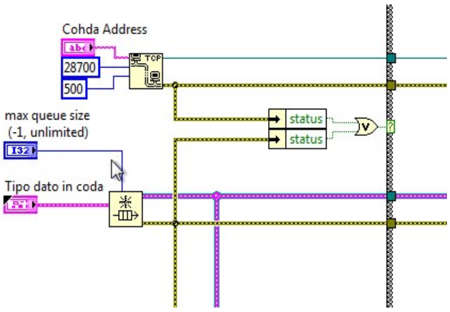


Figure 4.5: Connection handler between sbRIO and Cohda Wireless.

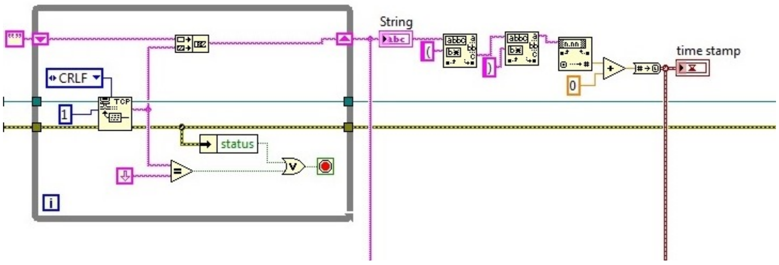


Figure 4.6: Extraction of data forwarded from the sbRIO.

Consumer task

The consumer organise messages collected inside the queue by the producer, according to their identifier: greater is the identifier, greater is the priority of the message. We remind that this criterion is strictly tied to the arbitration criterion of CAN bus. All these messages are then forwarded to the user

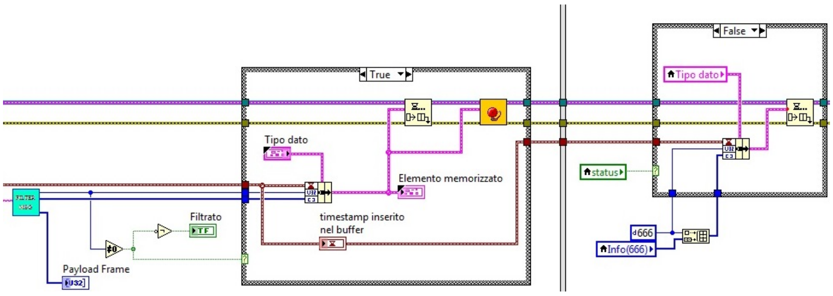


Figure 4.7: Filtering and enqueing of in-vehicle data.

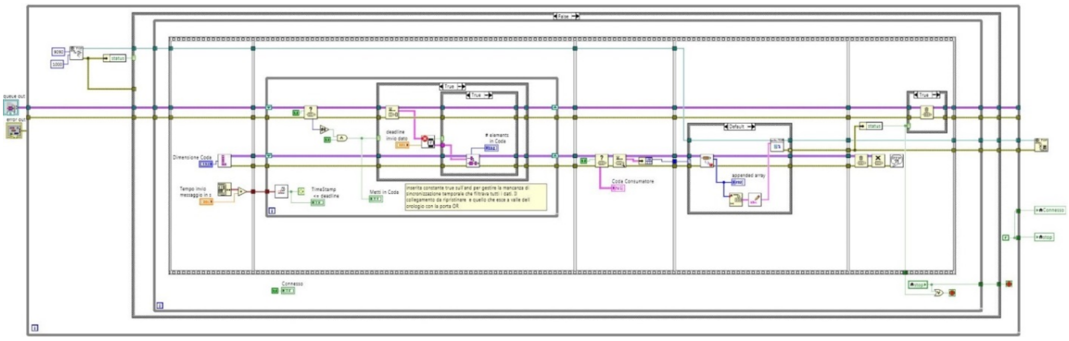


Figure 4.8: Consumer software structure.

interface via TCP/IP protocol. First of all the sbRIO listen for a connection, on a defined TCP port. If and only if the connection with the PDA is established, then the sort algorithm starts. The sorting algorithm takes care to reorder messages as function of their identifier and of their time-stamp: if a message is older with respect to a customizable time window, then is discarded and removed from the queue (see Fig. 4.9).

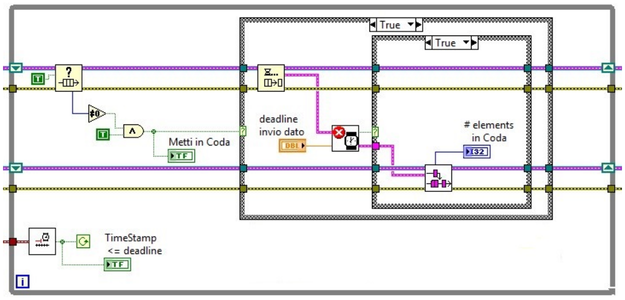


Figure 4.9: Consumer: sorting algorithm.

Data Collector task

This software module has the task to acquire and handles V2V data coming from neighbours vehicles. The received data will be computed, encapsulated and forwarded to the CACC algorithm, whose output is displayed as feedback, on the user interface. This mean that the user can behave as suggested on the user-interface, to bring the vehicle to the desired position with respect to the front one.

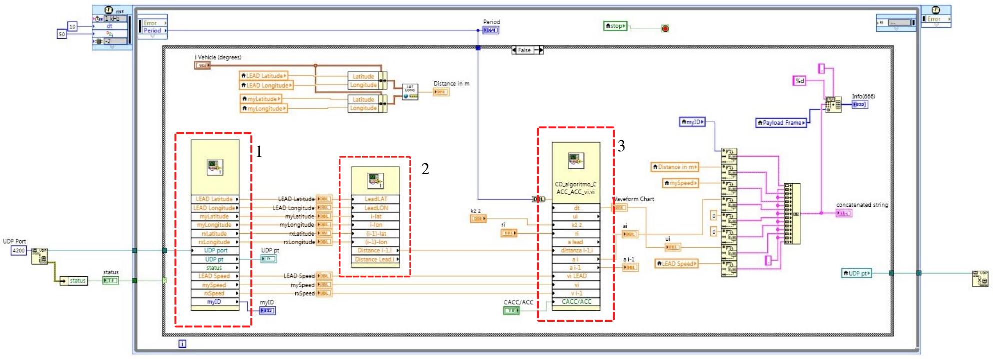


Figure 4.10: Consumer: data collector task.

The softaware architecture that has been designed, is mainly divided into 3 software blocks. The first software block, (from left to right in Fig. 4.10), fetches data from Cohda wireless: GPS coordinates, CAN data of the vehicle, and messages from neighbour vehicles. Then, a second software module, select properly data needed from the CACC algorithm, in particular position and speed of the preceding and leader vehicles providing as output the distance among i-th vehicle and the $i-1$-th. At this moment the distance among vehicles is computed with the haversine formula between two GPS point (see Equations (4.1)).

$$
\begin{aligned}
a &= \sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_1)\cos(\phi_2)\sin^2(\frac{\Delta\lambda}{2}) \\
c &= 2\mathrm{atan2}(\sqrt{\mathrm{a}}, \sqrt{(1\mathrm{a})}) \\
d &= Rc
\end{aligned}
\tag{4.1}
$$

With ϕ we indicate latitude converted into radians, with λ we indicate longitude converted into radians, while R is earths radius, $\Delta\phi = \phi_2 - phi_1$ and $\Delta\lambda = \lambda_2 - lambda_1$, d is the distance between the two coordinates. This approximation is sufficiently acceptable according to the desired spacing policy among vehicles that has been set to 30 m. The last software block, is responsible for the computation of the CACC algorithm. This algorithm has as input the speed and relative distance among vehicles and giving as output the desired set point of acceleration for the vehicle. This value represent the the control reference, starting from which, it will be possible to get the automatic actuation of vehicle.

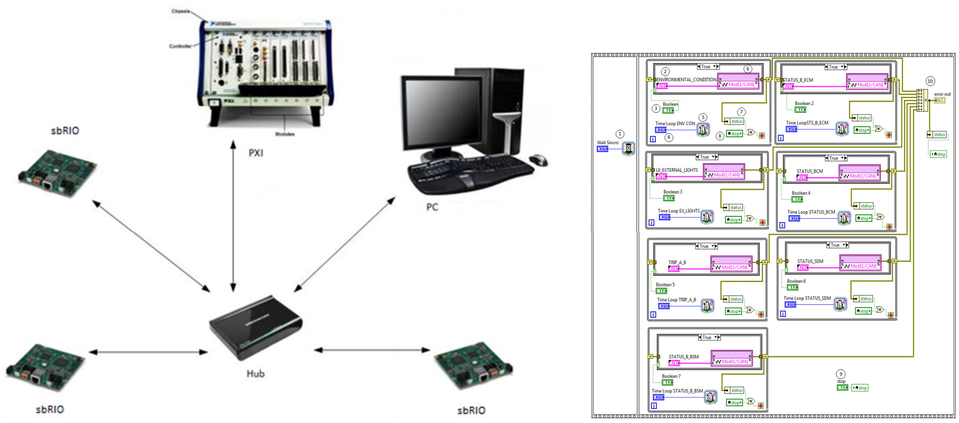
4.3 HIL Experimental setup

To validate the effectiveness of the proposed OBU's hardware/software architecture, HIL techniques have been used. The designed HIL environment is based on National Instruments devices, and is programmed into LabVIEW language to reproduce virtually the behaviour of vehicles, to test V2V algorithms, and of the remote system, to test V2I algorithms. The idea is to create a basic HIL platform in which would be possible to test and validate fleet's managing strategies and control algorithm deployed on the sbRIO. The hardware configuration of the HIL platform consists of:

- **PXI simulator.** This device is made up by of a modular NI PXIe-1062Q chassis with different slots. The core of the PXI system is the real time embedded controller module NI-PXIe-8135. that includes 2.3 GHz base frequency, 3.3 GHz (single-core Turbo Boost) quad-core processor, dual-channel 1600 MHz DDR3 memory.
- **3 sbRIO 9606.** See Sec. 4.2 for hardware characteristics.
- **PC Host.** Processor Intel i3 1.7 GHz, 4 GB memory, 500 GB hard drive.
- **Hub.** The 8 ports hub has been used to connect in a LAN network all the mentioned devices.

The overall architecture of the HIL platform is shown in Fig. 4.11a.

PXI simulator has been exploited to recreate a mock-up of the user-friendly graphical interfaces and to reproduce the the in-vehicle running condition. The mock-up of the user-interface, actually programmed on the Android mobile device, allow to emulate the presence of different users i.e. different OBUs inside a fleet. In this way is possible to generate a new fleet in the system, send V2I data to the an emulated remote server (we have used



(a) (b)

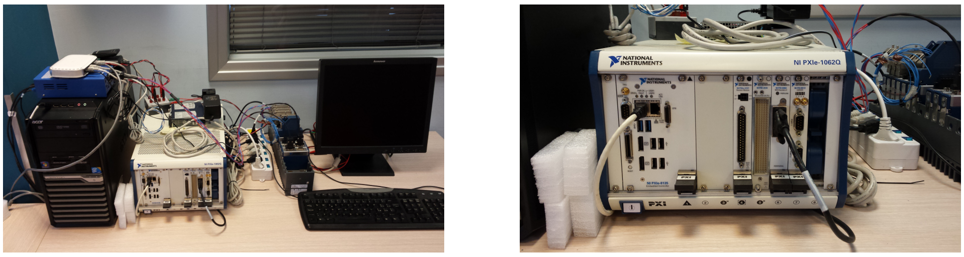
Figure 4.11: (a) HIL experimental setup. (b) FPGA LabVIEW design of Software ECU.

the PC Host) and send CAN data to other sbRIO. In the HIL setup, the 3 sbRIO emulate, in part, the OBUs behaviour. For this purpose, a software Electronic Control Unit (ECU) has been programmed exploiting PXI FPGA reconfigurable module. The software ECU reproduces all the inputs CAN frame to run the control and management algorithms deployed on the sbRIO (see Fig. 4.11b for detail). The PC host has been used for different tasks: *i*) it emulates the behaviour of a remote server that collect all data forwarded by vehicle in V2I communication. *ii*) It has been employed to manage, schedule all the hardware and software resources and to deploy the edited software on the real time devices (PXI and sbRIO). *iii*) Furthermore, a synoptic has been reproduced to resume the state of the overall architecture. We note that, the PXI, sbRIO and the PC communicate through the 5 port hub, forming a LAN of hosts each of which is characterized by an assigned IP address. Communication among devices has been programmed using the TCP/IP LabVIEW functions.

In Figs. 4.12a and 4.12b[1] is possible to see a detailed a detailed picture respectively of the HIL platform and of the PXI simulator that has been provided by Media Motive S.r.L. For detail on the experimental results we remands to [117].

Furthermore in Fig. 4.13 it is possible to note the screen of one laboratory test in HIL simulation, made in order to verify the correct acquisition of

[1] By courtesy of Media Motive S.r.L.

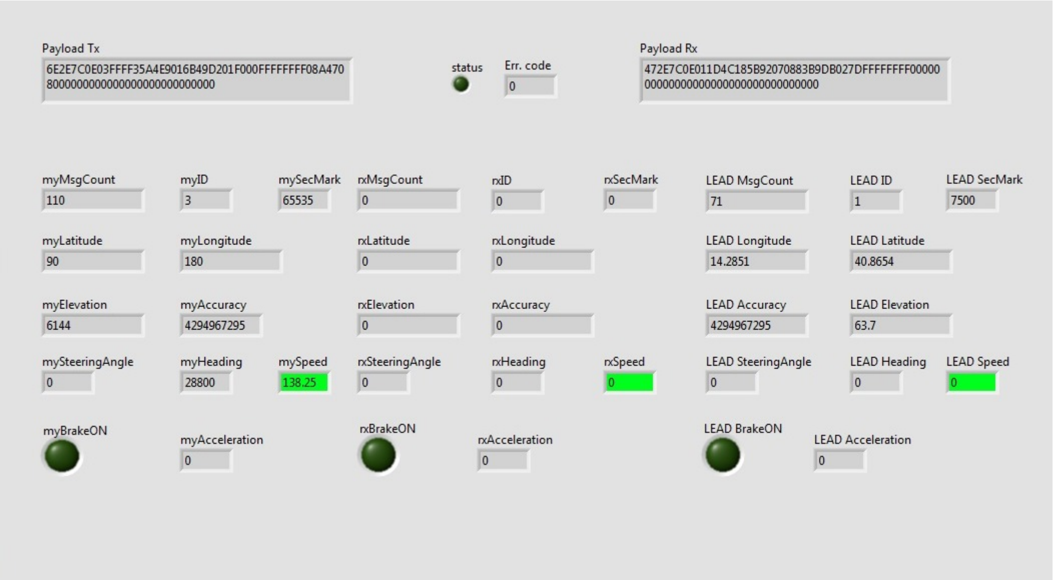


(a)

(b)

Figure 4.12: (a) Overall of the HIL platform. (b) Particular of the PXI simulator.

data within BSM messages. These messages are opportunely customized and used to implement the control strategy V2V.



Payload Tx
6E2E7C0E03FFFF35A4E9016B49D201F000FFFFFFFF08A470
800000000000000000000000000

status ●

Err. code 0

Payload Rx
472E7C0E011D4C185B92070883B9DB027DFFFFFFFF00000
000000000000000000000000000

myMsgCount 110
myID 3
mySecMark 65535
rxMsgCount 0
rxID 0
rxSecMark 0
LEAD MsgCount 71
LEAD ID 1
LEAD SecMark 7500

myLatitude 90
myLongitude 180
rxLatitude 0
rxLongitude 0
LEAD Longitude 14.2851
LEAD Latitude 40.8654

myElevation 6144
myAccuracy 4294967295
rxElevation 0
rxAccuracy 0
LEAD Accuracy 4294967295
LEAD Elevation 63.7

mySteeringAngle 0
myHeading 28800
mySpeed 138.25
rxSteeringAngle 0
rxHeading 0
rxSpeed 0
LEAD SteeringAngle 0
LEAD Heading 0
LEAD Speed 0

myBrakeON ●
myAcceleration 0
rxBrakeON ●
rxAcceleration 0
LEAD BrakeON ●
LEAD Acceleration 0

Figure 4.13: Testing in-vehicle.

Appendix A

Notation and Mathematical Preliminaries

Abstract

In this appendix we will cover the basic mathematical notation and theory exploited in this PhD thesis work. In particular we will focus on fundamentals of complex network theory, delayed system, Lyapunov-Krasovskii stability theorem, algebraic lemmas, matrix analysis and definitions.

A.1 Stability of Continuos-Time System

In what follow some useful results about stability of linear system are recalled. We begin by considering the familiar linear state equation:

$$\dot{x}(t) = Ax(t), \tag{A.1}$$

for this class of systems, the following result hold:

Theorem 4. *(Lyapunov stability for linear systems) Consider a linear system in the form of (A.1) and let $A \in \mathcal{R}^{N \times N}$. The following statements are equivalent:*

- *all the eigenvalues of A have negative real part;*
- *for all matrices $Q = Q^\top > 0$ there exists an unique solution $P = P^\top >$*

0 *to the following (Lyapunov) equation:*

$$AP^\top + PA = -Q \tag{A.2}$$

Consider now, a time varying linear system in the form:

$$\dot{x}(t) = A(t)x(t), \tag{A.3}$$

for $A \in \mathcal{R}^{N\times N}$ and $t \in \mathcal{R}$. Without loss of generality, assume that Eq. (A.3) has equilibrium $x = 0$. To establish asymptotic stability of this equilibrium, a standard approach is to seek standard quadratic Lyapunov function associated with Eq. (A.3). A typical choice is the classical quadratic function $V(x(t)) = x(t)^\top Px(t)$. Evaluating the time derivative $\dot{V}(t)$ along the system trajectory we have:

$$\dot{V}(x(t)) = x(t)^\top \left[A(t)^\top P + PA(t)\right] x(t), \tag{A.4}$$

and thus it is sufficient to seek a matrix $P \in \mathcal{S}_n^+$ which satisfies the continuous-time algebraic Lyapunov equation:

$$A(t)^\top P + PA(t) = -M(t), \tag{A.5}$$

where $M(t) \in \mathcal{S}_n^+$ is given. Here, $\mathcal{S}_n^+$ denotes the set of real, $n\times n$ positive definite symmetric matrices.

Theorem 5. *[27] The unique solution of Eq.* (A.5) *is given by:*

$$P(t) = \int_{t_0}^{\infty} \phi_A^\top(s,t_0)M(t)\phi_A(s,t_0)ds, \tag{A.6}$$

where $\phi_A(t,t_0)$ *is the transition matrix for the system Eq.* (A.3)*. Moreover,* $P \in \mathcal{S}_n^+$ *whenever* $M(t) \in \mathcal{S}_n^+$

On the other hand, suppose we seek a Lyapunov function of the form $V(x(t)) = x(t)^\top P(t)x(t)$, the emphasis being that P is time varying. Then

$$\dot{V}(x(t)) = x(t)^\top \left[A(t)^\top P(t) + P(t)A(t) + \dot{P}(t)\right] x(t), \tag{A.7}$$

and so we seek a $P(t) \in \mathcal{S}_n^+$ which satisfies the continuous-time differential

Lyapunov equation

$$A(t)^\top P(t) + P(t)A(t) + \dot{P}(t) = -M(t), \tag{A.8}$$

where $M(t) \in \mathcal{S}_n^+$ is specified.

Theorem 6. *[4, 27, 64] The unique solution of Eq.* (A.8)*, subject to the initial condition $P(t_0) = P_0$ is given by:*

$$P(t) = \phi_A^{-\top}(t,t_0)P(t_0)\phi_A^{-1}(t,t_0) - \int_{t_0}^{\infty} \phi_A^\top(s,t_0)M(t)\phi_A(s,t_0)ds \tag{A.9}$$

where $\phi_A(t,t_0)$ is the transition matrix for the system Eq. (A.3)*. Moreover, $P \in \mathcal{S}_n^+$ whenever $M(t) \in \mathcal{S}_n^+$*

A.2 Analysis of time delay systems

Time delay systems are systems in which a significant time delay exists between the applications of input to the system and their resulting effect. Such systems arise from an inherent time delay in the components of the system or from a deliberate introduction of time delay into the system for control purposes. Such time delay systems can be represented by delay differential equations, which belong to the class of functional differential equations [132].

The analysis of time-delay systems is a well-developed field gathering a lot of different techniques. These methods can be categorized to either belong to frequency-domain or time-domain techniques. Frequency-domain approaches are mostly devoted to linear time-invariant systems, yet under some circumstances, it is possible to adapt them to address the case of varying delays using, for instance, model transformations. Time-domain approaches can, however, be applied to any type of systems: linear or non-linear, with constant or time-varying delays, etc. [16]. Most of the existing results for stability of systems with time-varying delays, based on time-domain approaches, are developed based on the following two Lyapunov-type approaches [65]:

- The *Lyapunov-Razumikhin* method that looks for functions which normally allow one to prove stability of systems with bounded but freely fast time-varying delays. See for example papers [20] and [75].

- The *Lyapunov-Krasovskii* method that looks for functionals which only allow one to prove stability of time-delay systems where the delay parameters are bounded both in length and time variation ([39, 50, 66, 70, 82, 123]). In [39], a discussion about the conservatism among the different methods is given.

The main difference among the two approach relies int the fact that Razumikhin gives more conservative bound on the maximum allowable delay preserving stability than the Lyapunov-Krasovskii approach, as showed in [47].

A.2.1 Krasovskii theorem

In what follows we provide some definitions and results on the stability of delayed systems, according to Lyapunov-Krasovskii theory.

Definition 1. *(Uniform Norm) Denote $\mathcal{C}_{n,h} := \mathcal{C}([-h, 0], \mathcal{R}^n)$ as the Banach space of continuous vector functions mapping the interval $[-h, 0]$ to $\mathcal{R}^n$. Let $\phi(\theta) \in \mathcal{C}_{n,h}$, then the uniform norm of ϕ is defined as*

$$\|\phi\|_c = \sup_{\theta \in [-h,0]} \|\phi(\theta)\|$$

where $\|\cdot\|$ is the vector 2-norm.

where the vector norm $\|\cdot\|$ represents the $2-$norm $\|\cdot\|_2$. We use functional differential equations to describe time-delay systems. The general form of a Retarded Functional Differential Equation (RFDE) (or functional differential equation of retarded type) is:

$$\begin{array}{c} \dot{x}(t) = f(t, x_t),\ t \geq t_0 \\ x_{t_0}(\theta) = \phi(\theta),\ \theta \in [-h, 0] \end{array} \tag{A.10}$$

where $h > 0$, ϕ is the functional of initial conditions, $x_t(\cdot)$ denotes, for a given $t \geq t_0$, the restriction of $x(\cdot)$ to the interval $[t-h, t]$ translate to $[-h, 0]$, i.e.

$$x_t(\theta) = x(t + \theta)\ \forall \theta \in [-h, 0].$$

It is assumed that $\phi \in \mathcal{C}^l_{n,h} := \{\phi \in \mathcal{C}_{n,h} : \|\phi\|_c < l\}$, being l a positive real number, and the map $f(t, \phi) : \mathcal{R}^+ \times \mathcal{C}^l_{n,h} \to \mathcal{R}^n$ is continuous and Lipschitzian in ϕ and $f(t, 0) = 0$. Furthermore, we denote by $x(t; t_0, \phi)$ the solution of

(A.10) with the initial condition $(t_0, \phi) \in \mathcal{R}^+ \times \mathcal{C}^l_{n,h}$, generally referred to as the trivial solution. As in the study of systems without delay, an effective method for determining the stability of a time-delay system is Lyapunov method [46]. For a system without delay, this requires the construction of a Lyapunov function, which in some sense is a potential measure quantifying the deviation of the state $x(t)$ from the trivial solution 0 [46]. Since for a delay-free system $x(t)$ is needed to specify the systems future evolution beyond t, and since in a time-delay system the state at time t required for the same purpose is the value of $x(t)$ in the interval $[t-h, t]$, i.e., x_t , it is natural to expect that for a time-delay system, the corresponding Lyapunov function be a functional $V(t, x_t)$ depending on x_t, which also should measure the deviation of x_t from the trivial solution 0. Such a functional is known as a Lyapunov-Krasovskii functional [46]. For the class of systems in (A.10) it holds the following theorem:

Theorem 7. *(Lyapunov-Krasovskii Stability Theorem) [46] Suppose that the function $f : \mathcal{R} \times \mathcal{C}_{n,h} \to \mathcal{R}^n$ in (A.10) maps bounded set of $\mathcal{C}_{n,h}$ in bounded sets of $\mathcal{R}^n$, and suppose that $u(s)$, $v(s)$ and $w(s)$ are continuous non-negative and nondecreasing functions with $u(s)$, $v(s) > 0$ for $s \neq 0$ and $u(0) = v(0) = 0$. If there exists a continuous differentiable functional $V : \mathcal{R} \times \mathcal{C}_{n,h} \to \mathcal{R}$ such that*

$$u(\|\phi(0)\|) \leq V(t, \phi) \leq v(\| \phi\|_c) \tag{A.11}$$

and

$$\begin{aligned} \dot{V}(t, \phi) := & \lim_{\epsilon \to 0^+} \sup \frac{1}{\epsilon}[V(t+\epsilon, x_{t+\epsilon}(t_0, \phi)) - V(t, \phi)] \\ & \leq -w(\|\phi(0)\|), \end{aligned} \tag{A.12}$$

then the (trivial) solution $x = 0$ of (A.10) is uniformly stable. Moreover, if $w(s) > 0$ for $s > 0$, then it is uniformly asymptotically stable. In addition, if

$$\lim_{s \to \infty} u(s) = +\infty, \tag{A.13}$$

then it is globally uniformly asymptotically stable.

It is important to note that there exists two types of stability results, and can be be distinguished based on whether they depend on the delay value. In some cases it is possible to asses stability of delayed system for a range of delay values or even obtain stability results for family of delays. This leads us to the concepts of delay-independent and delay-dependent stability.

Definition 2. *(Delay-Independent Stability) [16] A time-delay system is stable independently of the delay or delay-independent stable if stability does not depend on the delay value, that is, if the system is stable for any delay value in* $[0, \infty]$.

The above definition immediately extends to systems with multiple delays and time-varying delays. This concept of stability is quite strong since delays must have no impact on stability. This imposes, in return, strong constraints on the structure of the system. It is therefore expected that time-delay systems are, most likely, not delay-independent stable.

Definition 3. *(Delay-Dependent Stability) [16] A time-delay system is delay-dependent stable if there exists a (bounded) interval $I \in \mathcal{R}$ for which the system is stable for any delay in I, and unstable otherwise.*

Unlike delay-independent stability, delay-dependent stability is a concept of stability that is actually sensitive to change in the delay values. This is certainly the most realistic notion of stability since delays are, most of the time, influential on the stability of real world systems.

Furthermore we recall here the definition of exponential stability in the case of time-delay systems:

Definition 4. *(Exponential stability of RFDE) [11] The trivial solution $x \equiv 0$ of* (A.10) *is said to be exponentially stable if there exists a $K \geq 0$ and a decay rate $\delta > 0$ such that for all initial conditions $\phi \in \mathcal{C}^l_{n,h}$ the solution satisfies the inequality*

$$\|x(t; t_0, \phi)\| \leq K e^{-\delta(t-t_0)} \|\phi\|_c \tag{A.14}$$

for all $t > t_0$.

A.2.2 Model transformation - Leibniz-Newton formula

Model transformation is a very common procedure introduced quite early in the analysis of time-delay systems, but not restricted to. The rationale behind model transformations is to turn a time-delay system into another system, referred to as a comparison system or comparison model, which may or may not be a time-delay system [16]. Analysis tools are then applied on the comparison system in order to draw conclusions on the stability of the original time-delay system. Model transformations lie at the core of many efficient analysis techniques such as Lyapunov-Razumikhin and Lyapunov-Krasovskii approaches. The goal of model transformations is to simplify the analysis of time-delay systems. The compensation for this is that the comparison system may exhibit additional dynamics leading to a possible loss of equivalence, in terms of stability, between the original and the comparison system. Additional dynamics consist of supplementary zeros in the characteristic equation of the comparison model. When at least one of these additional zeros is unstable, the comparison model is unstable and the stability of the original system cannot be inferred from the comparison model [46]. Many different model transformation procedures have been proposed in the literature, however, in this thesis work we will exploit the Leibniz-Newton formula [16].

Definition 5. *(Newton-Leibniz transformation) [16] The Newton-Leibniz model transformation is based on the following identity:*

$$x(t-h) = x(t) - \int_{t-h}^{t} \dot{x}(s)ds \tag{A.15}$$

A.3 Fundamentals of complex Networks

Traditionally control theory has focused on single systems and devices. New findings and applications (e.g. ITS), are challenging this traditional way of thinking. Nowadays a lot of researchers are interesting in control of more interacting systems to reach a common target: this problem is known as *control of complex networks*. In this case we speak about multi-agent system where with the term *agent*, we refer to an autonomous system, with an own dynamics, able to interact with surrounding environment and able to take decision to reach a specified target. The main goal in a multi-agent network is not to control the single system, but to control the whole network in order to produce a common behaviour applying distributed algorithms, to guarantee a smart group behaviour.

A.3.1 Network Modelling

Basically, to model a complex network we have to define:

1. A model of the dynamics of each agent;

2. The communication protocol (interaction model) between agents;

3. The structure of the interconnections between different agents.

Each agent can be thought of as a generic non-linear control system of the form:

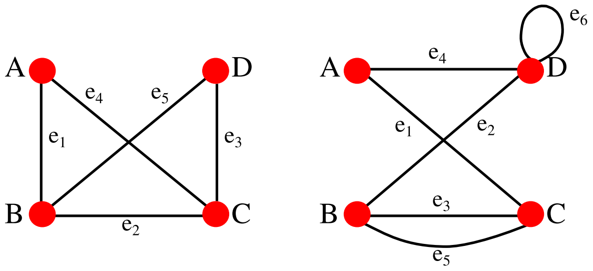
$$\dot{x}_i(t) = f_i(x_i(t)) + g_i(x_i(t))u_i(t), \tag{A.16}$$

with $x_i(t) \in \mathcal{R}^n$ and $u_i(t) \in \mathcal{R}^m$. The interaction between agents can be modelled by choosing an appropriate coupling law, e.g. the coupling between nodes can be modelled as:

$$u_i(t) = \sigma \sum_{j=1, j \neq i}^{N} \alpha_{i,j} h(x_i(t), x_j(t)), \tag{A.17}$$

where σ is the coupling gain, $\alpha_{i,j}$ model the presence/absence of coupling between agents in the network and $h(x_i(t), x_j(t))$ refers to the particular protocol used.

Usually the structure of the interconnections between different agents is modelled by what is commonly referred as a *network*. A network (also called a graph in the mathematical literature) is, a collection of vertices joined by edges. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a finite non empty set $\mathcal{V}$ of vertices and a finite set of edges, $\mathcal{E}$ [119]. The cardinality of $\mathcal{V}$ is called the order of $\mathcal{G}$. Some networks have a single edge between any pair of vertices. In the cases where there can be more than one edge between the same pair of vertices we refer to those edges collectively as a multi-edge. If there are also edges that connect vertices to themselves, we refer to those edges as self-edges or self-loops. A network that has neither self-edges nor multi-edges is called a simple network or simple graph. A network with multi-edges is called a multi-graph (Fig. A.1).



(a) Graph

(b) Multigraph

Figure A.1: Example of graphs: (a) simple graph; (b)multi-graph

There are different ways to characterize mathematically a network [45]. One of these representation is given by *adjacency matrix*. The adjacency matrix $\mathcal{A}$ is a matrix whose elements are 1 if and only if exists an edge from vertex i to vertex j. Consider an undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices, and let's label the vertices with integer labels $(1, ..., n)$. If we denote an edge between vertices i and j by (i, j) then the complete network can be specified by the matrix $\mathcal{A} \in \mathcal{R}^{n \times n}$, whose elements are so defined:

$$\alpha_{i,j} = \begin{cases} 1 & if : (i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}, \tag{A.18}$$

where $\mathcal{E}$ represent the set of the edges of graphs; while with $\mathcal{V}$ we denote the set of vertices. In some situations, however, it is useful to represent edges as having a strength, weight, or value to them, usually a real number. Thus in the Internet, for example, edges might have weights representing the amount of data flowing along them or their bandwidth. In this case we speak about weighted networks and the adjacency matrix will not have only elements equal to 1 or 0. Furthermore we can speak about delayed networks if the communication between agent i.e. each link of the network, is affected by a time-delay even though the agent is characterized by an own non delayed dynamics. To characterize analytically this situation, it is possible to define for each edge $(i, j) \in \mathcal{E}$ a function $\tau_{i,j}(t)$ that model the communication delay among the agent i nd the agent j. Indeed the assumption that there exist a communication time-delay between agent is a very realistic assumption for many real system as the World Wide Web. In reality the communication is not instantaneous, but the exchanged information is affected by a time-delay although sometimes negligible.

For every node $v_i \in \mathcal{V}$, we can define the set of neighbours N_i as the subset of $\mathcal{V}$ defined as follow:

$$N_i = \{j \in V : \alpha_{i,j} \neq 0\}. \tag{A.19}$$

The degree of a vertex in a graph is the number of edges connected to it. We will denote the degree of vertex i by Δ_i. In general Δ_i is so calculated:

$$\Delta_i = \sum_{j=1, j \neq i}^{N} \alpha_{i,j}. \tag{A.20}$$

Furthermore we can define the diagonal matrix $\Delta \in \mathcal{R}^{n \times n}$ whose diagonal element are the vertex degrees:

$$\Delta = \begin{bmatrix} \Delta_1 & 0 & ... & 0 \\ 0 & \Delta_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \Delta_n \end{bmatrix}. \tag{A.21}$$

Thanks to the degree matrix Δ and to the adjacency matrix $\mathcal{A}$, we can define an another important matrix, called Laplacian matrix $L \in \mathcal{R}^{N \times N}$.

$$L = \Delta - \mathcal{A} \tag{A.22}$$

For construction, the Laplacian matrix has zero row-sum, hence, at least one eigenvalue will be zero.

Digraph Propriety

A directed network or directed graph, also called a *digraph* for short, is a network in which each edge has a direction, pointing from one vertex to another. Such edges are themselves called directed edges, and can be represented by lines with arrows on them. Example of directed network is the World Wide Web, in which hyper-links run in one direction from one web page to another. Conversely a graph is defined undirected if each edge has not a direction. A digraph is strongly connected if there is a path from every node to every other node. A strong component of a digraph is an induced subgraph that is maximal, which is subject to being strongly connected. A directed tree is a digraph in which every node has exactly one parent node with the exception of one node, which is called the root, which has no parent and has a directed path to every other node. We say that j is reachable from node i if there exists a path from node i to node j. A node is said to be globally reachable if it is reachable from any other node in the graph. In the case of undirected graph the Laplacian matrix is a symmetric matrix with zero row-sum and real spectrum. For a digraph the Laplacian matrix is so defined:

$$L = [l_{i,j}] = \begin{cases} l_{i,i} = \displaystyle\sum_{j=1, j \neq i}^{N} \alpha_{i,j} & \\ l_{i,j} = -\alpha_{i,j} & i \neq j \end{cases} \tag{A.23}$$

For undirected graph,the Laplacian matrix, being symmetric, has eigenvalues that can be sorted as follow:

$$0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N \tag{A.24}$$

The smallest non-zero eigenvalue, λ_2 is positive if the network is connected and play an important role in the performance analyses of network. Before conclusion we recall the following lemma on global reachability of a node:

Lemma 4. *Digraph $\mathcal{G}_N$ has a globally reachable node if and only if the Laplacian of $\mathcal{G}_N$ has a simple zero eigenvalue (with eigenvector $\mathbf{1} = (1, ..., 1) \in \mathcal{R}^N$).*

A.3.2 Consensus and Synchronization in complex network

In networks of agents (or dynamic systems), consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. A consensus algorithm (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbours on the network [94]. Every agent exploit the same algorithm and take decision thanks to the local available information and those that receive from the other agents. Consider a network of agents interested in reaching a consensus via local communication with their neighbours on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that represent the agent connection. By reaching a consensus, we mean asymptotically converging to a one-dimensional agreement space characterized by the following equation:

$$\lim_{t \longrightarrow \infty} x_i(t) = x_i \ \forall i \in \mathcal{V} \tag{A.25}$$

This agreement space can be expressed as:

$$x = \alpha \mathbf{1}, \tag{A.26}$$

where $\mathbf{1} = (1, \cdots, 1)^\top$ and $\alpha \in \mathcal{R}$ is the collective decision of the group of agents. Related to consensus are synchronization phenomena arising in systems of coupled non-linear oscillators. To this aim the goal of synchronization can be analytically defined as follow:

$$\lim_{t \longrightarrow \infty} x_1(t) = ... = \lim_{t \longrightarrow \infty} x_n(t) = \lim_{t \longrightarrow \infty} s(t) \tag{A.27}$$

where $s(t)$ is a solution of an isolate node of the network $\dot{s}(t) = f(s(t))$. If (A.27) is satisfied all the agents present the same trajectory.

Consider a delayed complex network, the following definitions holds:

Definition 6. *(Synchronization in delay networks) [77, 137] Let $x_i(t, \phi)$ $(i = 1, 2, ..., N)$ be a solution of the i-th delayed system:*

$$\begin{array}{c} \dot{x}_i(t) = f_i(t, x_t),\ t \geq t_0 \\ x_i(t_0 + s) = \phi_i(s),\ s \in [-h, 0], \end{array} \tag{A.28}$$

and consider $\phi = [\phi_1, \phi_2, ..., \phi_N]^\top$ the vector of initial conditions. If there is a non empty subset $E \subseteq \Omega \subseteq \mathcal{R}^N$, with $\phi_i \in E(i = 1, ..., N)$, such that $x_i(t, \phi) \in \Omega$ for all $t \geq t_0$, $i = 1, 2, ..., N$, and

$$\lim_{t\to\infty} \|x_i(t, \phi) - s_0(t, \phi_0)\|_2 = 0 \ \ i = 1, 2, ..., N \tag{A.29}$$

where $\|\cdot\|$ is the Euclidean norm, $s_0(t, \phi_0)$ is an asymptotically stable solution of the system $\dot{s}_0 = f_0(s_0(t))$ with $s_0 \in \Omega$, then the dynamical networks (A.28) is said to realize synchronization and $E \times E \times \cdots \times E$ is called the region of synchrony for the dynamical networks (A.28) .

Definition 7. *(Synchronization manifold) [136] The hyperplane:*

$$\mathcal{S} = \left\{ \left[x_1^\top(t), x_2^\top(t), ..., x_N^\top(t)\right]^\top \in \mathcal{R}^{N\times N} : x_i(t) = x_j(t) = x_0(t) \right\}$$

for $i, j = 1, 2, ..., N$ is said to be the synchronization manifold of the delayed dynamical network (A.28), where $x_i(t) = [x_{i1}(t), x_{i2}(t), ..., x_{iN}(t)]^\top$ for $i = 1, 2, ..., N$ is the state of node i .

A.4 Matrix facts

In what follows we recall some linear algebra lemmas that will exploited in Chapter 3 for the convergence analysis of the proposed control strategy.

A.4.1 Extended Routh-Hurwitz criterion

First of all we recall the criterion of H. Bilharz, which is an extension of Hurwitz's criterion. According to this criterion, we recall an important result obtainable for polynomial with complex coefficients:

Lemma 5. *(Routh-Hurwitz criterion extension) [38] Assuming that a_0 is real and positive, the roots of a third-order polynomial with complex coefficients $\pi(z) = a_0 z^3 + (a_1 + ib_1)z^2 + (a_2 + ib_2)z + ... + (a_n + ib_n)$ are in the open left-half plane if and only if all "north-westerly" minors $D_1, D_2, ..., D_n$ of the Bilharz matrix (according to definition in [38]) are positive [97].*

Consider a third-order polynomial with complex coefficients as follow:

$$\pi(\lambda) = \left[a_0\lambda^3 + (a_1 + ib_1)\lambda^2 + (a_2 + ib_2)\lambda + (a_3 + ia_3)\right], \tag{A.30}$$

suppose that coefficient $a_0 = 1$. Its roots are in the open left-half plane if and only if the Bilharz matrix D_1 ,D_2, D_3, are positive definite. Where:

$$D_1 = a_1, \tag{A.31}$$

$$D_2 = \begin{pmatrix} a_1 & a_3 & -b_2 \\ a_0 & a_2 & -b_1 \\ 0 & b_2 & a_1 \end{pmatrix}, \tag{A.32}$$

and

$$D_3 = \begin{pmatrix} a_1 & a_3 & 0 & -b_2 & 0 \\ a_0 & a_2 & 0 & -b_1 & -b_3 \\ 0 & a_1 & a_3 & 0 & -b_2 \\ 0 & b_2 & 0 & a_1 & a_3 \\ 0 & b_1 & b_3 & a_0 & a_2 \end{pmatrix}. \tag{A.33}$$

A.4.2 Schur's Complements

Consider a square complex matrix given in the block form:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{A.34}$$

where A is a square matrix. If A is invertible, the Schur complement of A in M is so defined:

$$M/A = D - CA^{-1}B \tag{A.35}$$

Schur's complement plays an important role in various areas including matrix analysis, statistics, numerical analysis optimization. Among its numerous properties, perhaps, the most important and useful is the Schur determinant formula [60]:

$$det(M) = det(A)det(M/A) \tag{A.36}$$

A.5 Integral inequalities

In this section, some integral inequalities, exploited during the dissertation, have been retrieved.

First of all we recall the Hadamard inequality, valid for convex functions only:

Lemma 6. *(Hadamard Inequality) [33] Let $f : I \subseteq \mathcal{R} \to \mathcal{R}$ be a convex mapping defined on the interval I of real numbers, then the following inequality holds:*

$$\frac{1}{b-a}\int_a^b f(x)dx \leq \frac{f(a)+f(b)}{2}, \tag{A.37}$$

being $a, b \in I$ with $a < b$.

The following integral inequality is known as the Jensen Inequality, which plays an important role in the stability problem of time-delay systems:

Lemma 7. *(Jensen Inequality) [46] For any constant matrix $\Theta = \Theta^\top > 0 \in \mathcal{R}^{N\times N}$, scalar $h : h(t) > 0$, and vector function $x(.) : [-h, 0] \longrightarrow \mathcal{R}^n$ such that the following integral is defined, then*

$$h\int_{t-h}^{t} \eta^\top(s)\Theta\eta(s)ds \geqslant \int_{t-h}^{t} \eta^\top(s)ds\Theta\int_{t-h}^{t} \eta(s)ds. \tag{A.38}$$

Appendix B

Details on String Stability

The dynamics of the i-th vehicle can be recast in the s domain as

$$X_i(s) = H_i(s)U_i(s) + \frac{x_i(0)}{s^2} \tag{B.1}$$

with

$$X_i(s) = \mathcal{L}(r_i), U_i(s) = \mathcal{L}(u_i) H_i(s) = \frac{1}{T_i s^2 (s + \frac{1}{T_i})}$$

and being $x_i(0)$ the initial condition.

The Laplace transformation of the distributed coupling protocol Eq. (3.3) for $i = 1$ is:

$$U_1(s) = k_{10} E_1(s) + bs(X_0 - X_1) + \gamma s^2 (X_0 - X_1) \tag{B.2}$$

while, for $i = 2, ..., N$, we have:

$$\begin{aligned} U_i(s) = {} & \frac{k_{i0}}{\Delta_i} \left[s\tau^\star X_0 - X_i + X_0 e^{-\tau^\star s} + s h_{i0} X_0 + \frac{d_{i0}^{st}}{s} \right] \\ & + \frac{k_{i,i-1}}{\Delta_i} E_i + bs(X_0 - X_i) + \gamma s^2 (X_0 - X_i). \end{aligned} \tag{B.3}$$

being

$$E_i(s) = s\tau^\star X_0 - X_i + X_{i-1} e^{-\tau^\star s} + s h_{i,i-1} X_0 + \frac{d_{i,i-1}^{st}}{s} \tag{B.4}$$

the spacing error dynamics with respect to the preceding vehicle ($i = 1, ..., N$).

Substituting Eq. (B.3) in Eq. (B.1), after some algebraic manipulations, we obtain

$$\begin{aligned} X_i(s) = {} & H \frac{k_{i0}}{\Delta_i} \left[s\tau^\star X_0 - X_i + X_0 e^{-\tau^\star s} + s h_{i0} X_0 + \frac{d_{i0}^{st}}{s} \right] \\ & + H \frac{k_{i,i-1}}{\Delta_i} E_i + bHs(X_0 - X_i) + \gamma H s^2 (X_0 - X_i) + \frac{x_i(0)}{s^2}. \end{aligned} \tag{B.5}$$

The spacing error can be now computed in terms of the sensitivity functions $T_i(s)$ and $S_i(s)$ $(i = 2, .., N)$ as

$$E_i(s) = T_i(s)E_{i-1}(s) + S_i(s)\frac{d^{st}_{i,i-1}}{s} \tag{B.6}$$

where:

$$\begin{aligned}
&T_i = \frac{1}{(-1 - D_i)}[C_i + (C_i - e^{-\tau^\star s})k_{i0}\bar{H} + F_i +\\
&+(C_i - e^{-\tau^\star s})(b\bar{H}s + \gamma s^2\bar{H})]W^{-1}_{i-1};\\
&\\
&S_i = \frac{1}{(-1 - D_i)}[F_i + (C_i - e^{-\tau^\star s})(b\bar{H}s + \gamma s^2\bar{H})]W^{-1}_{i-1}S_{i-1} +\\
&+\frac{(C_i - e^{-\tau^\star s})}{1 + b\bar{H}s + s^2\gamma\bar{H}} - 1 + C_i + \frac{2}{B_i}\frac{1}{s}]
\end{aligned} \tag{B.7}$$

with

$$\begin{aligned}
D_i &= H\frac{k_{i,i-1}}{\Delta_i}\frac{1}{B_i}\\
B_i &= 1 + H\frac{k_{i0}}{\Delta_i} + sbH + s^2\gamma H\\
C_i &= H\frac{k_{i0}}{\Delta_i}\frac{1}{B_i}\\
\bar{H} &= \frac{H}{1 + bHs + \gamma Hs^2}\\
F_i &= \left(-\tau^\star s + \frac{bHs}{B_i} + \frac{\gamma Hs^2}{B_2} + C_2 sh_{i,i-1} - sh_{i,i-1}\right)\\
W_i &= \frac{(e^{-\tau^\star} + \tau^\star s) - b\bar{H}s - s^2\gamma\bar{H} + sh_{i,0}}{\bar{H}k_{i0} + 1}.
\end{aligned} \tag{B.8}$$

Bibliography

[1] DSRC Implementation Guide - A guide to users of SAE J273 5 message sets over DSRC. Std 20, SAE International, Feb 2010.

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

[3] IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation. Std 1609.4, IEEE, February 2011.

[4] Hisham Abou-Kandil. *Matrix Riccati equations: in control and systems theory*. Springer, 2003.

[5] Waleed Alasmary and Weihua Zhuang. Mobility impact in IEEE 802.11 p infrastructureless vehicular networks. *Ad Hoc Networks*, 10(2):222–230, 2012.

[6] Luis Alvarez and Roberto Horowitz. Safe platooning in automated highway systems. *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[7] Chang Ande, Jiang Guiyan, and Niu Shifeng. Traffic congestion identification method based on gps equipped floating car. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 3, pages 1069–1071. IEEE, 2010.

[8] Oreste Andrisano, Roberto Verdone, and Masao Nakagawa. Intelligent transportation systems: the role of third generation mobile radio networks. *Communications Magazine, IEEE*, 38(9):144–151, 2000.

[9] L.D. Baskar, B. De Schutter, J. Hellendoorn, and Z. Papp. Traffic control and intelligent vehicle highway systems: a survey. *Intelligent Transport Systems, IET*, 5(1):38–52, March 2011.

[10] Sagar Behere, Martin Törngren, and De-Jiu Chen. A reference architecture for cooperative driving. *journal of Systems Architecture*, 59(10):1095–1112, 2013.

[11] Richard Ernest Bellman and Kenneth L Cooke. Differential-difference equations. 1963.

[12] Selim Benhimane, Ezio Malis, Patrick Rives, and José R Azinheira. Vision-based control for car platooning using homography decomposition. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2161–2166. IEEE, 2005.

[13] R. Bishop. A survey of intelligent vehicle applications worldwide. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 25–30, 2000.

[14] Gayatry Borboruah and Gypsy Nandi. A study on large scale network simulators5.

[15] Alessio Botta, Antonio Pescapè, and Giorgio Ventre. Quality of Service Statistics over Heterogeneous Networks: Analysis and Applications. *European Journal of Operational Research*, 191(3):1075–1088, 2008.

[16] Corentin Briat. *Linear Parameter-Varying and Time-Delay Systems*. Springer, 2014.

[17] Fanping Bu, Han-Shue Tan, and Jihua Huang. Design and field testing of a cooperative adaptive cruise control system. In *American Control Conference (ACC), 2010*, pages 4616–4621. IEEE, 2010.

[18] Wilco Burghout. *Hybrid microscopic-mesoscopic traffic simulation*. PhD thesis, Royal Institute of Technology Stockholm, Sweden, 2004.

[19] Andreea Elena Blu. *Overall powertrain modeling and control based on driveline subsystems integration*. PhD thesis, 2011.

[20] Yong-Yan Cao, You-Xian Sun, and Chuwang Cheng. Delay-dependent robust stabilization of uncertain systems with multiple state delays. *Automatic Control, IEEE Transactions on*, 43(11):1608–1612, 1998.

[21] D. Caveney. Cooperative vehicular safety applications. *Control Systems, IEEE*, 30(4):38–53, Aug 2010.

[22] Derek Caveney and William B Dunbar. Cooperative driving: beyond v2v as an adas sensor. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 529–534. IEEE, 2012.

[23] Gang Chen and Frank L. Lewis. Leader-following Control for Multiple Inertial Agents. *International Journal of Robust and Nonlinear Control*, 21(8):925 – 942, 2011.

[24] E. Coelingh and S. Solyom. All Aboard the Robotic Road Train. *IEEE Spectrum*, 49(11):34 – 39, 2012.

[25] Erik Coelingh and Stefan Solyom. All aboard the robotic road train. *Spectrum, IEEE*, 49(11):34–39, 2012.

[26] Yan Cui. $L2 - L_\infty$ Consensus Control for High-Order Multi-Agent Systems with Nonuniform Time-Varying Delays. *Asian Journal of Control*, 2014.

[27] John M Davis, Ian A Gravagne, Robert J Marks, and Alice A Ramos. Algebraic and dynamic lyapunov equations on time scales. In *System Theory (SSST), 2010 42nd Southeastern Symposium on*, pages 329–334. IEEE, 2010.

[28] A.J. Day. *Braking of Road Vehicles*. Elsevier Science, 2014.

[29] Luigi Del Re, Frank Allgöwer, Luigi Glielmo, Carlos Guardiola, and Ilya Kolmanovsky. *Automotive model predictive control: models, methods and applications*, volume 402. Springer, 2010.

[30] MiXiM developers. Original design by Andreas Viklund. Mixim website. Available at ¡http://mixim.sourceforge.net/¿, 2014.

[31] M. di Bernardo, P. Falcone, A. Salvi, and S. Santini. Design, analysis and experimental validation of a distributed protocol for platooning

in the presence of time-varying heterogeneous delays. *submitted for publication to IEEE Transactions on Control Systems Technology*, 2015.

[32] M. di Bernardo, A. Salvi, and S. Santini. Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time Varying Heterogeneous Communication Delays. *to appear in IEEE Trans. on Intelligent Transportation Systems*, 2014.

[33] SS Dragomir, J Pecaric, and LE Persson. Some inequalities of hadamard type. *Soochow J. Math*, 21(3):335–341, 1995.

[34] D. Eckhoff, C. Sommer, and F. Dressler. On the Necessity of Accurate IEEE 802.11P Models for IVC Protocol Simulation. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–5, May 2012.

[35] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *Bell System Technical Journal*, pages 1538–7305, 1963.

[36] Pedro Fernandes and Urbano Nunes. Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):91–106, 2012.

[37] L. Figueiredo, I. Jesus, J.A.T. Machado, J.R. Ferreira, and J.L.M. de Carvalho. Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 1206–1211, 2001.

[38] Evelyn Frank. On the zeros of polynomials with complex coefficients. *Bulletin of the American Mathematical Society*, 52(2):144–157, 1946.

[39] E Fridman and Uri Shaked. Delay-dependent stability and h control: constant and time-varying delays. *International journal of control*, 76(1):48–60, 2003.

[40] Emilia Fridman and Yury Orlov. Exponential stability of linear distributed parameter systems with time-varying delays. *Automatica*, 45(1):194–201, 2009.

[41] Sae Fujii, Atsushi Fujita, Takaaki Umedu, Shigeru Kaneda, Hirozumi Yamaguchi, T Higashinoz, and Mineo Takai. Cooperative vehicle positioning via v2v communications and onboard sensors. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011.

[42] J.C. Gerdes and J.K. Hedrick. Vehicle speed and spacing control via coordinated throttle and brake actuation. *Control Engineering Practice*, 5(11):1607 – 1614, 1997.

[43] E. N. Gilbert. Capacity of a Burst-Noise Channel. *Bell System Technical Journal*, page 12531265, 1960.

[44] Thomas D Gillespie. Fundamentals of vehicle dynamics. Technical report, SAE Technical Paper, 1992.

[45] Christopher David Godsil, Gordon Royle, and CD Godsil. *Algebraic graph theory*, volume 207. Springer New York, 2001.

[46] K. Gu, V.L. Kharitonov, and J. Chen. *Stability of Time-Delay Systems*. Control Engineering. Birkhäuser Boston, 2012.

[47] Keqin Gu and Silviu-Iulian Niculescu. Survey on recent results in the stability and control of time-delay systems*. *Journal of Dynamic Systems, Measurement, and Control*, 125(2):158–165, 2003.

[48] Suraj G Gupta, Mangesh M Ghonge, Parag D Thakare, and PM Jawandhiya. Open-source network simulation tools: An overview. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(4):pp–1629, 2013.

[49] Jack K Hale and Sjoerd M Verduyn Lunel. *Introduction to Functional Differential Equations*. Applied mathematical science. Springer-Verlag New York Inc., 1993.

[50] Qing-Long Han and Keqin Gu. Stability of linear systems with time-varying delay: a generalized discretized lyapunov functional approach. *Asian Journal of Control*, 3(3):170–180, 2001.

[51] He Hao and Prabir Barooah. Stability and Robustness of Large Platoons of Vehicles with Double-integrator Models and Nearest Neighbor Interaction. *International Journal of Robust and Nonlinear Control*, pages 1099 – 1125, 2012.

[52] H. Hartenstein and K. Laberteaux. *VANET Vehicular Applications and Inter-Networking Technologies*. Intelligent Transport Systems. Wiley, 2009.

[53] Gerhard Hasslinger and Oliver Hohlfeld. The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference*, pages 1–15, March 2008.

[54] Ping He, Chun-Guo Jing, Tao Fan, and Chang-Zhong Chen. Robust decentralized adaptive synchronization of general complex networks with coupling delayed and uncertainties. *Complexity*, 19(3):10–26, 2014.

[55] Ping He, Chun-Guo Jing, Tao Fan, and Chang-Zhong Chen. Robust decentralized adaptive synchronization of general complex networks with coupling delayed and uncertainties. *Complexity*, 19(3):10–26, 2014.

[56] JK Hedrick, Masayoshi Tomizuka, and P Varaiya. Control issues in automated highway systems. *Control Systems, IEEE*, 14(6):21–32, 1994.

[57] Andrea Heide and Klaus Henning. The cognitive car: A roadmap for research issues in the automotive sector. *Annual reviews in control*, 30(2):197–203, 2006.

[58] LV Hien, Quang Phuc Ha, and Vu Ngoc Phat. Stability and stabilization of switched linear dynamic systems with time delay and uncertainties. *Applied Mathematics and Computation*, 210(1):223–231, 2009.

[59] Jean-Michel Hoc, Mark S Young, and Jean-Marc Blosseville. Cooperation between drivers and automation: implications for safety. *Theoretical Issues in Ergonomics Science*, 10(2):135–160, 2009.

[60] R. A. Horn and C. R. Johnson. *Matrix Analisis*. University Press, Cambridge, 1987.

[61] Jiangping Hu and Yiguang Hong. Leader-following coordination of multi-agent systems with coupling time delays. *Physica A: Statistical Mechanics and its Applications*, 374(2):853 – 863, 2007.

[62] Weng Jian-cheng, Zhai Ya-qiao, Zhao Xiao-juan, and Rong Jian. Floating Car Data Based Taxi Operation Characteristics Analysis in Beijing. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 5, pages 508–512, March 2009.

[63] Fangcui Jiang, Long Wang, and Guangming Xie. Consensus of high-order dynamic multi-agent systems with switching topology and time-varying delays. *Journal of Control Theory and Applications*, 8(1):52–60, 2010.

[64] Rudolf E Kalman and John E Bertram. Control system analysis and design via the second method of lyapunov: Icontinuous-time systems. *Journal of Fluids Engineering*, 82(2):371–393, 1960.

[65] Chung-Yao Kaoa and Bo Lincolnb. Simple stability criteria for systems with time-varying delays. *Automatica*, 40:1429–1434, 2004.

[66] Vladimir L Kharitonov and S-I Niculescu. On the stability of linear systems with uncertain delay. *Automatic Control, IEEE Transactions on*, 48(1):127–132, 2003.

[67] Roozbeh Kianfar, Bruno Augusto, Alireza Ebadighajari, Usman Hakeem, Josef Nilsson, Ali Raza, Reza S Tabar, NV Irukulapati, Christer Englund, Paolo Falcone, et al. Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):994–1007, 2012.

[68] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline, and Vehicle.* Springer, 2005.

[69] Maria Kihl. Vehicular network applications and services. *Vehicular Networks: Techniques, Standards, and Applications*, 2009.

[70] Jin-Hoon Kim. Delay and its time-derivative dependent robust stability of time-delayed linear systems with uncertainty. *Automatic Control, IEEE Transactions on*, 46(5):789–792, 2001.

[71] YI Kyongsu, WOO Minsu, KIM Sunga, and LEE Seongchul. A study on a road-adaptive cw/ca algorithm for automobiles using hil simulations. *JSME International Journal Series C*, 42(1):163–170, 1999.

[72] Li Li and Fei-Yue Wang. Cooperative driving at blind crossings using intervehicle communication. *Vehicular Technology, IEEE Transactions on*, 55(6):1712–1724, 2006.

[73] Li Li, Ding Wen, Nan-Ning Zheng, and Lin-Cheng Shen. Cognitive cars: A new frontier for adas research. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):395–407, March 2012.

[74] Perry Li, Luis Alvarez, Roberto Horowitz, Pin-Yen Chen, and Jason Carbaugh. A safe velocity tracking controller for ahs platoon leaders. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, volume 2, pages 2283–2288. IEEE, 1996.

[75] Xi Li and Carlos E De Souza. Criteria for robust stability and stabilization of uncertain linear systems with state delay. *Automatica*, 33(9):1657–1662, 1997.

[76] Xiangheng Liu, A Goldsmith, S.S. Mahal, and J.K. Hedrick. Effects of Communication Delay on String Stability in Vehicle Platoons. In *IEEE Intelligent Transportation Systems Conference (ITSC 2001)*, pages 625–630, 2001.

[77] Jinhu Lu and Guanrong Chen. A time-varying complex dynamical network model and its controlled synchronization criteria. *Automatic Control, IEEE Transactions on*, 50(6):841–846, 2005.

[78] J. Erdmann M. Behrisch, L. Bieker and D. Krajzewicz. Sumo simulation of urban mobility. In *SIMUL 2011 : The Third International Conference on Advances in System Simulation*, pages 55–60, 2011.

[79] Pietro Marchetta, Alessandro Salvi, Eduard Natale, Antonio Tirri, Manuela Tufo, and Davide De Pasquale. S 2-move: Smart and social move. In *Global Information Infrastructure and Networking Symposium (GIIS), 2012*, pages 1–6. IEEE, 2012.

[80] Jan Philipp Maschuw and Dirk Abel. Longitudinal vehicle guidance in networks with changing communication topology. In *Advances in Automotive Control*, pages 785–790, 2010.

[81] Barbara M Masini, Cesare Fontana, and Roberto Verdone. Provision of an emergency warning service through gprs: performance evaluation.

In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 1098–1102. IEEE, 2004.

[82] D Mehdi, EK Boukas, and Zi-Kuan Liu. Dynamical systems with multiple time-varying delays: Stability and stabilizability. *Journal of optimization theory and applications*, 113(3):537–565, 2002.

[83] Vicente Milanés, Steven E. Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative Adaptive Cruise Control in Real Traffic Situations. *IEEE Trans. on Intelligent Transportation Systems*, 15(1):296–305, February 2014.

[84] Christoph Sommer 2006-2012 Last modified: 2014-02-24. VEINS. Available at ¡http://veins.car2x.org/¿, 2014.

[85] Umberto Montanaro, Manuela Tufo, Giovanni Fiengo, and Stefania Santini. A novel cooperative adaptive cruise control approach: Theory and hardware in the loop experimental validation. In *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, pages 37–42. IEEE, 2014.

[86] Seungwuk Moon, Ilki Moon, and Kyongsu Yi. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Engineering Practice*, 17(4):442–455, 2009.

[87] Hassnaa Moustafa, Sidi Mohammed Senouci, and Moez Jerbi. Introduction to vehicular networks. *Vehicular Networks*, page 1, 2009.

[88] Hassnaa Moustafa and Yan Zhang. *Vehicular networks: techniques, standards, and applications.* Auerbach publications, 2009.

[89] Richard M Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[90] Todd Murray, M Cojocari, and Huirong Fu. Measuring the Performance of IEEE 802.11p Using NS-2 Simulator for Vehicular Networks. In *IEEE EIT*, pp. 498 – 503, 2008.

[91] G.J.L. Naus, R.P.A. Vugts, J. Ploeg, M.J.G. van de Molengraft, and M. Steinbuch. String-Stable CACC Design and Experimental Vali-

dation: A Frequency-Domain Approach. *IEEE Trans. on Vehicular Technology*, 59(9):4268 – 4279, 2010.

[92] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[93] H. Noori and B.B. Olyaei. A novel study on beaconing for VANET-based vehicle to vehicle communication: Probability of beacon delivery in realistic large-scale urban area using 802.11p. In *Smart Communications in Network Technologies (SaCoNeT), 2013 International Conference on*, volume 01, pages 1–6, June 2013.

[94] Reza Olfati-Saber, J. Alex Fax, and Richard Murray. Consensus and Cooperation in Networked Multi-Agent System. In *Proceedings of the IEEE*, volume 95, pages 215–233, January 2007.

[95] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.

[96] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95, November 2009.

[97] C. Parks and V. Hahn. *Stability Theory*. Ellis Horwood Books in Information Technology. Prentice Hall, 1993.

[98] J. Ploeg, B.T.M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer. Design and Experimental Evaluation of Cooperative Adaptive Cruise Control. In *ITSC 2011*, pages 260–265. IEEE, Oct. 2011.

[99] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

[100] R. Rajamani. *Vehicle Dynamics and Control*. Springer, second edition, 2012.

[101] Nedžad Repčić, Isad Šarić, and Vahid Avdić. Tractive effort curves in gearbox analyse.

[102] Research and Innovative Technology Administration (RITA). Intelligent transportation system joint program office. Available at ¡http://www.its.dot.gov/¿, 2014.

[103] Michele Segata, Stefan Joerer, Bastian Bloessl, Christoph Sommer, Falko Dressler, and Renato Lo Cigno. PLEXE: A Platooning Extension for Veins. In *IEEE VNC*, Paderborn, Germany, Dec 2014.

[104] Michele Segata and Renato Lo Cigno. Automatic Emergency Braking - Realistic Analysis of Car Dynamics and Network Performance. *IEEE Transactions on Vehicular Technology*, 62(9):4150–4161, October 2013.

[105] A Seuret and F. Gouaisbaut. Integral inequality for time-varying delay systems. In *Control Conference (ECC), 2013 European*, pages 3360–3365, July 2013.

[106] Elaine Shaw and J Karl Hedrick. String stability analysis for heterogeneous vehicle strings. In *American Control Conference, 2007. ACC'07*, pages 3118–3125. IEEE, 2007.

[107] S.E. Shladover. Path at 20 history and major milestones. *Intelligent Transportation Systems, IEEE Transactions on*, 8(4):584–592, Dec 2007.

[108] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Trans. on Mobile Computing*, 10(1):3–15, 2011.

[109] Karl R Stromberg. *An Introduction to Classical Real Analysis*. Wadsworth International, Belmont, 1981.

[110] D. Swaroop and J.K. Hedrick. String Stability of Interconnected Systems. *IEEE Trans. on Automatic Control*, 41(3):349 – 357, 1996.

[111] Car to Car Communication Consortium et al. C2c-cc manifesto, 2007.

[112] Masayoshi Tomizuka. Automated highway systems-an intelligent transportation system for the next century. In *Advanced Intelligent Mechatronics' 97. Final Program and Abstracts., IEEE/ASME International Conference on*, page 1. IEEE, 1997.

[113] Peng-Jui Tseng, Chia-Chen Hung, Tsung-Hsun Chang, and Yu-Hsiang Chuang. Real-time urban traffic sensing with gps equipped probe vehicles. In *ITS Telecommunications (ITST), 2012 12th International Conference on*, pages 306–310. IEEE, 2012.

[114] Sadayuki Tsugawa, Shin Kato, Kiyohito Tokuda, Takeshi Matsui, and Haruki Fujii. A cooperative driving system with automated vehicles and inter-vehicle communications in demo 2000. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 918–923. IEEE, 2001.

[115] Sasayuki Tsugawa, Shin Kato, Takeshi Matsui, Hiroshi Naganawa, and H Fujii. An architecture for cooperative driving of automated vehicles. In *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, pages 422–427. IEEE, 2000.

[116] A. Vahidi and A. Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *Intelligent Transportation Systems, IEEE Transactions on*, 4(3):143–153, Sept 2003.

[117] A.S. Valente, U. Montanaro, M. Tufo, A. Salvi, and S. Santini. Design of a platoon management strategy and its hardware-in-the loop validation. In *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, pages 1–5, May 2014.

[118] Rob ECM Van der Heijden and Vincent AWJ Marchau. Advanced driver assistance systems: behavioural implications of some recent developments. *European Journal of Transport and Infrastructure Research 5 (4), 239-252.(2005)*, 2005.

[119] Remco Van Der Hofstad. Random graphs and complex networks. *Available on http://www. win. tue. nl/rhofstad/NotesRGCN. pdf*, 2009.

[120] M. van Eenennaam, A. van de Venis, and G. Karagiannis. Impact of ieee 1609.4 channel switching on the ieee 802.11p beaconing performance. In *Wireless Days (WD), 2012 IFIP*, pages 1–8, Nov 2012.

[121] Wim Vandenberghe, Erik Vanhauwaert, Sofie Verbrugge, Ingrid Moerman, and Piet Demeester. Feasibility of expanding traffic monitoring

systems with floating car data technology. *IET Intelligent Transport Systems*, 6(4):347–354, 2012.

[122] Pravin Varaiya. Smart cars on smart roads: problems of control. *Automatic Control, IEEE Transactions on*, 38(2):195–207, 1993.

[123] Erik I Verriest. Robust stability of time varying systems with unknown bounded delays. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 1, pages 417–422. IEEE, 1994.

[124] Ruben Visser, Ir CJG van Driel, Ir HH Versteegt, and City Enschede. Co-operative driving on highways, 2005.

[125] Pangwei Wang, Yunpeng Wang, Guizhen Yu, and Tieqiao Tang. An improved cooperative adaptive cruise control (cacc) algorithm considering invalid communication. *Chinese Journal of Mechanical Engineering*, 27(3):468–474, 2014.

[126] Zengyun Wang, Lihong Huang, and Yaonan Wang. Robust decentralized adaptive control for a class of uncertain neural networks with time-varying delays. *Applied Mathematics and Computation*, 215(12):4154–4163, 2010.

[127] Ding Wen, Gongjun Yan, Nan-Ning Zheng, Lin-Cheng Shen, and Li Li. Toward cognitive vehicles. *Intelligent Systems, IEEE*, 26(3):76–80, 2011.

[128] Jo Yung Wong. *Theory of ground vehicles.* John Wiley & Sons, 2001.

[129] Shi Xiaohui, Xing Jianping, Zhang Jun, Zhou Lei, and Li Weiye. Application of dynamic traffic flow map by using real time gps data equipped vehicles. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 1191–1194. IEEE, 2006.

[130] Wen Yang, A.L. Bertozzi, and Xiaofan Wang. Stability of a Second Order Consensus Algorithm with Time Delay. In *47th IEEE Conference on Decision and Control (CDC)*, pages 2926–2931, 2008.

[131] Her-Terng Yau and Jun-Juh Yan. Robust decentralized adaptive control for uncertain large-scale delayed systems with input nonlinearities. *Chaos, Solitons & Fractals*, 39(4):1515–1521, 2009.

[132] Sun Yi, Yi Sun, et al. *Time-delay systems: analysis and control using the Lambert W function.* World Scientific, 2010.

[133] Xunqi Yu, J.W. Modestino, and Xusheng Tian. The accuracy of Gilbert models in predicting packet-loss statistics for a single-multiplexer network model. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2602–2612 vol. 4, March 2005.

[134] Huaguang Zhang, Mo Zhao, Zhiliang Wang, and Zhenning Wu. Adaptive synchronization of an uncertain coupling complex network with time-delay. *Nonlinear Dynamics*, pages 1–11, 2014.

[135] Wei-Song Zhong, Georgi M Dimirovski, and Jun Zhao. Adaptive synchronization for a class of complex delayed dynamical networks. In *American Control Conference*, pages 5121–5126, 2008.

[136] Jin Zhou and Tianping Chen. Synchronization in general complex delayed dynamical networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(3):733–744, 2006.

[137] Jin Zhou, Junan Lu, and Jinhu Lu. Adaptive synchronization of an uncertain complex dynamical network. *arXiv preprint nlin/0512031*, 2005.

[138] Wei Zhu and Daizhan Cheng. Leader-following consensus of second-order agents with multiple time-varying delays. *Automatica*, 46(12):1994 – 1999, 2010.

List of Figures

List of Tables