PLEXE: A Platooning Extension for Veins

Michele Segata*†, Stefan Joerer*, Bastian Bloessl‡, Christoph Sommer‡, Falko Dressler*‡, Renato Lo Cigno†
*Computer and Communication Systems, Institute of Computer Science, University of Innsbruck, Austria
†Dept. of Information Engineering and Computer Science, University of Trento, Italy
‡Distributed Embedded Systems, Dept. of Computer Science, University of Paderborn, Germany
{segata,joerer,bloessl,sommer,dressler}@ccs-labs.org,
locigno@disi.unitn.it

***Abstract*—Cooperative driving in general and Cooperative Adaptive Cruise Control (CACC) or platooning in particular require blending control theory, communications and networking, as well as mechanics and physics. Given the lack of an integrated modeling framework and theory as well as the prohibitively high costs of using prototypes for what-if studies, simulation remains the fundamental instrument to evaluate entire cooperative driving systems. This work presents PLEXE, an Open Source extension to Veins that offers researchers a simulation environment able to run experiments in realistic scenarios, taking into account physics and mechanics of the vehicles, communications and networking impairments, and Inter-Vehicle Communication (IVC) protocol stacks. PLEXE is easily extensible and already implements protocols to support platooning and cooperative driving applications and several state of the art cruise control models. We describe the structure of the simulator and the control algorithms that PLEXE implements and provide two use cases which show the potential of our framework as a powerful research tool for cooperative driving systems.**

I. INTRODUCTION

In the quest to improve road infrastructure usage and to boost safety, cooperative driving with active information exchange between vehicles using Inter-Vehicle Communication (IVC) has been identified as a key solution, if not the most important one. The challenge is to develop vehicles that relieve humans from the driving duty, in an efficient, smart, and safe way. Modern cars already provide the so called Adaptive Cruise Control (ACC), i.e., a radar-based system which automatically maintains a safety gap to the vehicle in front. Such system however does not improve road traffic efficiency because of the size of the safety gaps it has to maintain. For this reason, projects like PATH and SARTRE started to work on its evolution towards Cooperative Adaptive Cruise Control (CACC) [1], [2].

CACC exploits IVC to enhance system's reaction time, thus providing improved safety even while driving at small distances. Most projects employ Dedicated Short Range Communications (DSRC) for IVC, but some consider LTE/LTE-A as well, all focusing on very low latency real-time communication among the involved vehicles.

The overall challenge does not merely include CACC design, but all the procedures and maneuvers that will enable the so called *platooning* application. In this work, the term platooning indicates general cooperative driving applications that involves platoons, i.e., groups of vehicles that coordinate their movement. Applications include emergency braking, multi-vehicle automated driving and maneuvering, and not only vehicle-following applications. Indeed, designing a CACC means answering the question *"How do we make vehicles automatically follow each other?"*, but the questions that platooning arise are many more: *"How do we form, manage, and disrupt platoons?"*, *"Can the network support crowded scenarios?"*, *"How does a platoon interact with human-driven vehicles?"*. So far, these questions have only partially been answered.

We can find theoretical [3], experimental [4]–[6], and simulative studies (or a combination of them) on platooning. Simulative studies provide the basis for showing characteristics of designed controllers [7], performing large scale analysis, investigating maneuvers and possible interactions with human-driven vehicles [8], or proving benefits on traffic flows [9]. The limitations of simulative studies is that they are often missing important aspects (e.g., communication impairments or vehicles' physics), or the experiments lack reproducibility because the simulator is not available or it is proprietary. This is especially worrying, because the tools used to draw the scientific conclusions are not available for peer review.

In this paper, we present PLEXE[1], an extension to the well known and widely used Veins[2] simulation framework [10], introducing several enhanced capabilities that enable realistic studies of platooning concepts and applications. Making use of the capabilities of Veins to simulate both the communication among vehicles as well as their mobility within the road network, PLEXE also integrates all the necessary components to study platooning ranging from controller models to maneuvers to form and to maintain platoons.

The contributions of the paper can be summarized as follows:

- We provide the community with a free, Open Source tool that can be used for testing platooning control models and maneuvers in large-scale and mixed scenarios;
- We describe the structure of PLEXE and detail the cruise control models it implements to ease the process of customizing it;
- We provide two use cases that show the potential of the simulator. In particular, one example extends PLEXE by implementing a user-defined control algorithm and by comparing it with the ones provided by the simulator, while the second shows how to use PLEXE to implement and analyze a join maneuver.

[1] http://plexe.car2x.org [2] http://veins.car2x.org

II. Background and Related Work

The aim of this section is to reason about the new integrated approach used in PLEXE. We concentrate on the works that developed or used a platooning simulation framework for performance analysis. Thus, we focus on simulation environments that allow investigation of the joint performance of road and telecommunication networks, finally listing the features that a platooning simulation tool should have.

A. Simulation Tools

Fernandes and Nunes [11] developed a platooning simulator based on extensions to SUMO, implementing a CACC car following model, which was described in [5]. The simulator, however, is not publicly available and assumes all vehicles but the leader to be CACC driven, thus it is not possible to consider legacy vehicles, and the tool assumes a simple scenario. The road is a single-lane highway, so formation/disruption of platoons cannot be investigated, as well as any other maneuver. The authors focus essentially on the vehicle dynamics perspective, assuming a synchronized slotted communication protocol where no interference, collisions, and packet losses occur.

The goal of the work presented in [12] is developing a simulator, called Hestia, for the vehicular part of the problem and for testing different solutions based on a distributed agent modelling. It is implemented in a 3D environment to be able to capture all aspects concerning vehicle dynamics, down to object detection through sensors, as well as engine and road conditions. The communication part instead is idealized, without a proper simulation of the communication network at the packet and signal levels. Moreover, it does not support mixed scenarios and its scaling properties are not discussed.

van Arem et al. [9] studied the impact of CACC systems on traffic flow. To reach their goals, they developed a stochastic simulation model, which features a multi-lane highway where vehicles can change lanes and overtake each other. Moreover, different vehicle types can be used, and the simulator can emulate both human and automated behaviors. However, the simulator does not consider communication impairments, packets are never lost and channel interference is not considered.

The work by Lei et al. [7] performs a simulative study of a CACC system in order to determine the effects of packet losses on the string-stability of the controller. The authors develop a complex system where control laws are implemented in Matlab/SIMULINK, road network and vehicles are handled by SUMO, and the network is simulated by OMNeT++. The simulator is however dedicated to the analysis of the specific control system and not publicly available.

On the other hand, studies like [13] are more interested in realistic network simulation rather than mobility, thus use a simulator which implements a full IEEE 802.11p stack and provides realistic fading and shadowing phenomena.

The authors of [14] take one step toward a complete simulator by extending Veins [10], as we do. The simulator is tightly tailored to the purposes of the paper, where the Intelligent Driver Model (IDM) is considered, thus featuring no CACC-like controllers. Also, the source code is not publicly available.

Another advanced simulator from a vehicle dynamics point of view is presented by Zhao et al. in [15]. The simulator is an extension of the commercial tool VISSIM and features a human-behavioral model, together with ACC and CACC controllers, and already implements some platooning management maneuvers, thus enabling studies of mixed scenarios with platoon formations and disruptions. The communication network, however, is not implemented and simulated, and the tool is not available to the community.

We ourselves started working on the concept of platooning simulation using the vehicular networking simulator Veins [16]. Yet, this earlier work did not consider the integration of all the needed controller and maneuver models.

B. Numerical Analyses

The authors of [8] tackle the problem of human-driven vehicles interference by proposing a self-defensive maneuvering strategy. To evaluate their proposal, they develop a simulator in Matlab/SIMULINK, but no details for vehicle or networking models are given. Similarly, the simulator developed and used in [17] is based on solving differential equations and focuses on the vehicle dynamics aspect, while the authors of [18] perform the evaluation of the proposed CACC using numerical analysis. As another example, di Bernardo et al. [19] develop a new idea for a cooperative controller and they first analyze its stability properties in theory, and then confirm their results by means of Matlab simulations, but no vehicles or communications are actually taken into account.

C. Towards an Integrated Approach

The simulators listed in this section provide the ground for the identification of the requirements for a fully-featured platooning simulator, which are:

- Openness: It must be available online, free to download and to modify according to specific purposes;
- Active maintenance: It must be kept up-to-date and it must improve with time. To this purpose, it is a good idea to start from a widely used Vehicular Ad Hoc Network (VANET) simulator which will give a solid base, and the openness policy will permit the community to identify potential bugs and improve it;
- Realistic simulation of wireless and road networks: Realism of network simulation is of uttermost importance when analyzing these systems, as they heavily rely on communications in order to work. On the other hand, vehicle dynamics simulation is as important as the network to understand how a vehicle would behave in a real world environment;
- Extensibility: Users must easily be able to create new traffic scenarios or include new vehicle control models;
- Mixed traffic: As the introduction of CACC systems will be gradual, it is important to be able to simulate scenarios where automated and human-driven vehicles coexist.

These are the features that characterize PLEXE, the Open Source platooning simulator we present in this work.

III. Simulator Structure

Veins extends the OMNeT++ network simulator [20] by providing a complete vehicular communication stack based on IEEE 802.11p and tailored channel models, together with a way of modeling realistic node mobility based on the road traffic simulator SUMO [21]. For this it couples the network and the mobility simulator by creating a network node in OMNeT++ for each vehicle travelling in SUMO. Each OMNeT++ node is 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. The coupling between the network and the traffic simulation frameworks is done through the TraCI interface which SUMO exposes. 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 is travelling on, or its acceleration.

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 Veins can be fed to the CACCs in SUMO. Platooning 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. Figure 1 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 maneuvers 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. The version of PLEXE we present in this work is based on Veins 2.2 and SUMO 0.17.1, and it will be constantly upgraded to follow the evolution of both frameworks.

A. Implementing Platooning Capabilities in SUMO

The implementation mainly refers to a new framework for car-following, 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 described in Section IV and generically called Cruise Control (CC) available and accessible via TraCI.

SUMO car-following models are conceived to mimic the behavior of drivers. Common examples are the IDM [22], or the Krauss [23] car following models. The car-following model we developed is called CC, which stands for Cruise Controllers. The user can modify the model to implement other

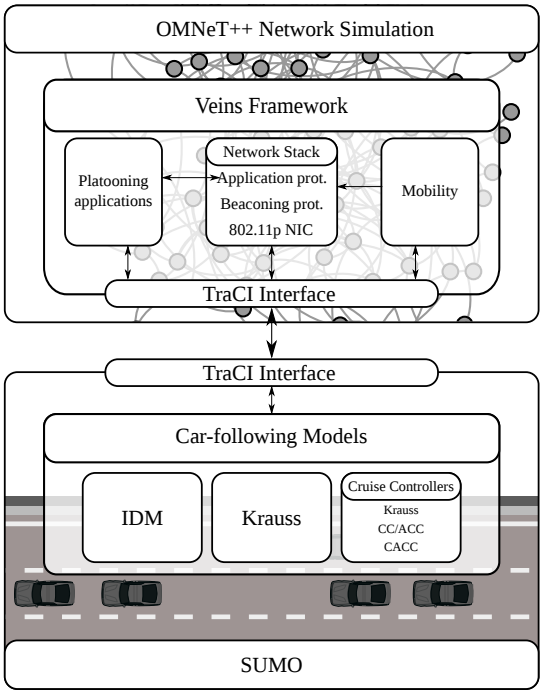


Figure 1. Schematic structure of the simulator.

CC or CACC models[3]. By default, CC uses the Krauss model to drive the car. This is 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. 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 in SUMO now includes standard CC/ACC and one advanced CACC, plus the engine actuation lag (please refer to Section IV for the details), while another advanced CACC is under testing and will be available soon. Through the TraCI interface it is possible to access the model, changing its behavior and retrieving different information. For the complete list of functionalities accessible via TraCI, please refer to the simulator documentation, here we just describe the most important ones. Concerning the CCs, it is possible to set desired speed $\dot{x}_{\text{des}}$, headway time T, and desired distance gap_{des}. 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 [19].

At runtime, the user can choose which component is controlling the car, i.e., the human behavioral model, any of the CC, ACC, or CACC, or any other controlling model implemented in the simulator by the user itself. For testing and safety verification purposes, it is possible to set different

[3] For a detailed description of the code and examples on how to modify it, please refer to the online documentation available at http://plexe.car2x.org.

behaviors for a platoon leader, e.g., a constant acceleration. For instance, to simulate an emergency braking, we can set leader's deceleration to $6\,\mathrm{m/s^2}$ until it comes to a complete stop. Other functionalities include setting a fixed lane to travel on, or retrieving distance to route end for example.

B. Platooning Protocols and Applications in Veins

Concerning Veins, besides the required changes to the TraCI interface, we provide a basic network 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 base class, named `BaseProtocol`, provides functionalities to inheriting classes like logging of statistics, primitives for sending and receiving packets, and loading of simulation parameters. This way subclasses can focus just on the implementation of the beaconing strategy itself.

The same principle applies to the application layer, where `BaseApp` takes care of loading simulation parameters, or passing data to the CACC via TraCI. Platooning applications are in charge, for instance, of deciding if a particular car is the leader of the platoon or not, on which lane a car should travel, if a car has to join or leave a platoon, and so on. The highest layer provides primitives to implement platooning applications, simplifying their description and implementation. Samples of already available primitives and platooning applications beyond car-following are given in Section V.

This stack is provided as a basic example that researchers can use as a starting point, but it is possible to develop a brand-new structure and simply use the functionalities provided by the CC car-following model in SUMO through the TraCI interface.

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.

IV. Controller Models

In vehicles the independent control parameter for longitudinal motion is the desired acceleration $\ddot{x}_{\text{des}}$ *actuated* by the car through the throttle control (if the engine or brakes permit). The actuation is not immediate because of the lag induced by power-train dynamics. In PLEXE the *actuation lag* is modelled by a first order low-pass filter [24, Chapter 5], which means that the actual acceleration applied to the car is computed as

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

$$\beta = \frac{\Delta_t}{\tau + \Delta_t} \tag{2}$$

The acceleration at simulation step n is computed based on the desired acceleration (computed by the controller) and the acceleration in the previous simulation step. τ is the time constant (default set to $0.5\,\mathrm{s}$), while Δ_t is the SUMO update step in seconds. The acceleration is in any case limited to model physical limits: $\ddot{x} \in [a_{\min}; a_{\max}]$.

Controllers can be implemented based on these fundamental physics of motion control. The first one that PLEXE provides is the classic Cruise Control (CC) [24, Chapter 5], already available on several commercial cars, which allows the driver to select a desired speed automatically maintained by the vehicle. The control law is defined as

$$\ddot{x}_{\text{des}} = -k_p \left(\dot{x} - \dot{x}_{\text{des}}\right) + \eta \tag{3}$$

where $\dot{x}$ is the current speed, $\dot{x}_{\text{des}}$ is the desired speed. k_p is the gain of the proportional controller (default set to 1); η is a random disturbance taking into account imprecisions of the actuator and of the speed measure (default set to 0).

Since the only inputs to the CC are the desired and actual speed, to avoid a collision the driver needs to manually disable the CC when approaching a slower vehicle. To relieve the driver from this duty, high-end cars now include a radar or laser scanner as well. The system detects slower vehicles, decelerates, and automatically maintains a safe distance. This is known as Adaptive Cruise Control (ACC) [24, Chapter 6] and its control law is defined as

$$\ddot{x}_{i\_des} = -\frac{1}{T}\left(\dot{\varepsilon}_i + \lambda \delta_i\right) \tag{4}$$

$$\delta_i = x_i - x_{i-1} + l_{i-1} + T\dot{x}_i \tag{5}$$

$$\dot{\varepsilon}_i = \dot{x}_i - \dot{x}_{i-1} \tag{6}$$

where i identifies the controlled vehicle and $i-1$ the vehicle in front. T is the time headway in seconds, $\dot{\varepsilon}_i$ is the relative speed between vehicle i and $i-1$, l_{i-1} is the length of the vehicle in front and δ_i is the distance error, i.e., the difference between the actual distance ($x_i - x_{i-1} + l_{i-1}$) and the desired distance ($T\dot{x}_i$). The distance $T\dot{x}_i$ grows proportionally with speed, and for both safety and stability reasons the time headway $T > 2 \cdot \tau$ (see [24] for further details). λ is a design parameter strictly greater than 0 (default set to 0.1).

When the ACC is selected, the interaction between CC and ACC is implemented as $\ddot{x}_{\text{des}} = \min(\ddot{x}_{\text{CC}}, \ddot{x}_{\text{ACC}})$. Basically, if the CC is mandating to accelerate to reach the desired speed, but the ACC is mandating to slow down because of a vehicle in front, the car follows the instructions of the ACC. Conversely, if the ACC is mandating to accelerate to follow the car in front, but the car has reached its desired speed, the CC will make the car "detach" from the one in front. Moreover, if there is no vehicle in front or the distance is higher than $250\,\mathrm{m}$, then the model considers only the CC, assuming that the radar detects no car in front.

The controllers introduced so far are useful to complement human behavioral driving models already present in SUMO, and to control a platoon leader. However, they are not suited for followers, since the requirements for a platooning system are small inter-vehicle distance and *string stability*. A platooning controller is string stable when it is able to attenuate the propagation of motion disturbances at the head of the platoon toward the tail of the platoon. The class of controllers able to realize platoon driving is known as Cooperative Adaptive

Cruise Control (CACC). CACCs exploit data that vehicles share among each other by means of IVC. PLEXE includes two different CACCs, thus it is an ideal environment to introduce new ones and compare them with the state of the art.

The first CACC is taken from [24, Chapter 7] and it is based on classical control theory. It uses data from the leading and preceding vehicle, and it is capable to maintain a fixed, speed-independent inter-vehicle distance.

The control law of the i-th vehicle in the platoon is

$$\ddot{x}_{i\_des} = \alpha_1 \ddot{x}_{i-1} + \alpha_2 \ddot{x}_0 + \alpha_3 \dot{\varepsilon}_i + \alpha_4 \left(\dot{x}_i - \dot{x}_0\right) + \alpha_5 \varepsilon_i \quad (7)$$

$$\varepsilon_i = x_i - x_{i-1} + l_{i-1} + \text{gap}_{\text{des}} \quad (8)$$

$$\dot{\varepsilon}_i = \dot{x}_i - \dot{x}_{i-1} \quad (9)$$

$\ddot{x}_0$ and $\dot{x}_0$ are the acceleration and speed of the leader respectively, while $\ddot{x}_{i-1}$ is the acceleration of the preceding vehicle. The distance error ε_i is based on a constant desired distance gap_{des} in meters (5 m by default).

The α_i parameters in Equation (7) are

$$\alpha_1 = 1 - C_1; \quad \alpha_2 = C_1; \quad \alpha_5 = -\omega_n^2 \quad (10)$$

$$\alpha_3 = -\left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)\omega_n \quad (11)$$

$$\alpha_4 = -C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n. \quad (12)$$

C_1 is a weighting factor between the accelerations of the leader and the preceding vehicle (default set to 0.5), ξ is the damping ratio (default set to 1), and ω_n is the bandwidth of the controller (default set to 0.2 Hz); defaults are taken from [25].

The interaction of the CACC with the CC is performed depending on the distance. If a vehicle is farther than 20 m from the front one, the policy is the same as for ACC: $\ddot{x}_{\text{des}} = \min(\ddot{x}_{\text{CC}}, \ddot{x}_{\text{CACC}})$, otherwise $\ddot{x}_{\text{des}} = \ddot{x}_{\text{CACC}}$. This way it is possible to have two different maximum accelerations, $a_{\text{max,CC}}$ for the CC (limited for comfort reasons) and the absolute maximum and minimum a_{max} and a_{min} representing the vehicle's limit.

Currently we are implementing a CACC inspired by [19] and based on the theory of consensus (the code will be published when it is stable). This is a completely different control model, but thanks to the decoupling of the vehicles' dynamics from the control algorithms and from the protocols implementation, the integration of a new control model is easy. We plan to run a comparison of the two algorithms soon, which would be one of the very first comparative studies in platooning.

V. PLATOON MANEUVERING

Platooning is not only about automated car following and string stability. Platoons needs to be created, maintained, modified and disrupted. A platoon needs to cope with interfering vehicles, and it may need to overtake a slower vehicle (or platoon). To make it short, a platoon needs to be able to behave as a single "flexible" vehicle driven by an intelligent professional driver. The study, the development, and the testing of maneuvers are important parts of this field [17], [26]–[28], and PLEXE provides a framework for the development and testing of maneuvers and their supporting protocols.

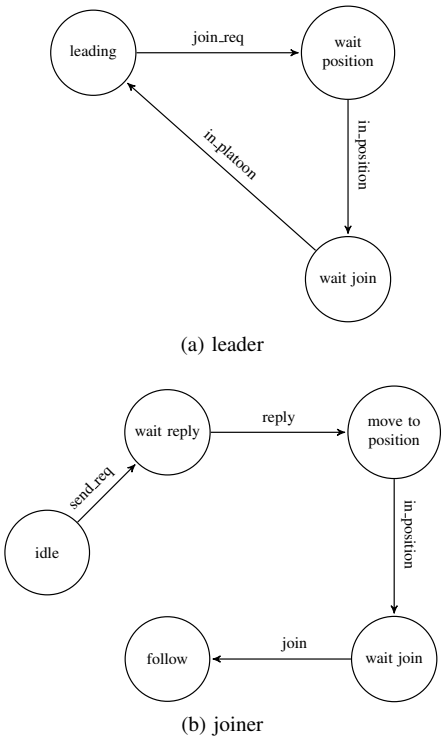


Figure 2. State machines of the sample join maneuver.

The architecture we choose to support these features is based on providing protocol primitives (elementary messages like join request/response) as well as basic application building blocks that use the primitives to implement simple maneuvers. We explain this approach by means of an example for the sake of clarity; the interested reader can find more details, primitives and elementary maneuvers in PLEXE documentation, and the performance of more complex maneuvers in [28].

Consider a car that approaches a platoon and wants to join at the tail. Figure 2 shows the state machines of the protocol that leading and joining vehicles use in order to coordinate the maneuver. The other vehicles are not actively involved, but are informed overhearing the message exchange, so that the platooning management can take the appropriate measures and, for instance, inform drivers of the additional vehicle. The leading and joining vehicles start from the LEADING and IDLE states respectively. The joiner requests the leader to join the platoon with the `send_req` primitive ("join platoon" is a parameter of the primitive), and moves to the WAIT REPLY state. The leader answers with the `join_req` primitive, which relays information about the platoon (lane, join position, etc.), and moves to the WAIT POSITION state. The joiner uses such data to come close to the tail of the platoon, moving into the join position. Once in position, i.e., at a negotiated distance from the last car of the platoon, the joiner notifies the leader that it is able to join. The leader sends back a confirmation and the joiner switches to CACC, closing its gap to the predecessor to the platoon inter-car distance. At the end of the procedure, the leader switches back to the LEADING state, while the joiner to FOLLOW. The source code of this example can

Table I
NETWORK AND ROAD TRAFFIC SIMULATION PARAMETERS.

	Parameter	Value
communication	Path loss model	Free space ($\alpha = 2.0$)
	PHY model	IEEE 802.11p
	MAC model	1609.4 single channel (CCH)
	Frequency	5.89 GHz
	Bitrate	6 Mbit/s (QPSK R = $1/2$)
	Access category	AC_VI
	MSDU size	200 B
	Transmit power	20 dBm
mobility	Leader's average speed	100 km/h
	Oscillation frequency	0.2 Hz
	Oscillation amplitude	$\simeq$ 95 km/h to 105 km/h
	Platoon size	8 cars
	Car length	4 m
controllers	Engine lag τ	0.5 s
	Weighting factor C_1	0.5
	Controller bandwidth ω_n	0.2 Hz
	Damping factor ξ	1
	Desired gap gap_{des}	5 m
	Headway time T	0.3 s and 1.2 s
	ACC parameter λ	0.1
	Distance gain k_d	0.7
	Speed gain k_s	1.0
	Desired speed $\dot{x}_{\text{des}}$ (followers)	130 km/h

be found under the `plexe-1.1-join-example` branch for Veins. The changes to the TraCI interface that PLEXE provides are sufficient to support these maneuvers, which can be implemented in Veins without further SUMO modifications.

VI. USE CASES

In this section we provide two sample use cases for the simulator to demonstrate its versatility. In particular, in the first example we implement a new fictional controller and we compare it with the ACC and CACC control algorithms provided by the simulator. In the second one, we implement a join maneuver, i.e., a car approaching and joining a platoon of four cars travelling on the same freeway. The results presented refer to a single run for demonstration purpose. Runs can be repeated to obtain statistical confidence.

A. Controller Analysis

We start by describing the fictional controller we implement for demonstration purpose. For the sake of brevity, we do not detail all steps that are necessary to implement the user-defined controller. Such details can be found in the documentation, while the source code is available in the `git` repository under the `plexe-1.1-mycc-example` branch, both for Veins and for SUMO.

We define the control law of the custom CACC as

$$\ddot{x}_{i\_\text{des}} = k_d\left(x_{i-1} - x_i - l_{i-1} - 25\,\text{m}\right) + k_s\left(\dot{x}_{i-1} - \dot{x}_i\right) \quad (13)$$

We refer to this controller with TESTCC. TESTCC aims at maintaining an inter-vehicle distance of 25 m and the same speed of the vehicle in front, and has two design gains k_d and k_s. The radar provides the distance to the front vehicle, while the speed is obtained by means of wireless communication.

We test the performance of TESTCC against ACC and CACC (Equations (4) and (7) respectively). We run a platoon of eight cars on a stretch of freeway. The platoon is leaded by a car which continuously changes its speed in a sinusoidal fashion. Table I summarizes communication, mobility, and controllers parameters for the simulations.

Figure 3 shows the speed profile of the vehicles in time for the different controllers. Thickest and darkest lines represent vehicles at the head of the platoon, while thinnest and lightest the ones at the tail.

The first comparison concerns ACC with two different headway times T. Figure 3a shows the behavior of the ACC algorithm when violating the string stability constraint, i.e., when setting $T = 0.3$ s. Follower vehicles are not able to attenuate leader-induced disturbance resulting in an increasing speed amplitude toward the end of the platoon. Conversely, when setting $T = 1.2$ s (Figure 3b), the speed oscillation is progressively attenuated by the vehicles. We can however observe a tracking lag by looking at how the speed of one vehicle is out of phase with respect to the one of its predecessor. Both phenomenons are reduced to the minimum when using the CACC (Figure 3c). Thanks to the controller design, each car can perfectly track leader's behavior by using its acceleration and speed. Finally, Figure 3d shows the poor design of TESTCC. The algorithm is definitely unstable, and vehicles at the end of the platoon amplify the disturbances in an uncontrolled manner. As shown, an oscillation of leader's speed between 95 km/h and 105 km/h results in speed range exceeding 80 km/h and 120 km/h for some vehicles at the tail.

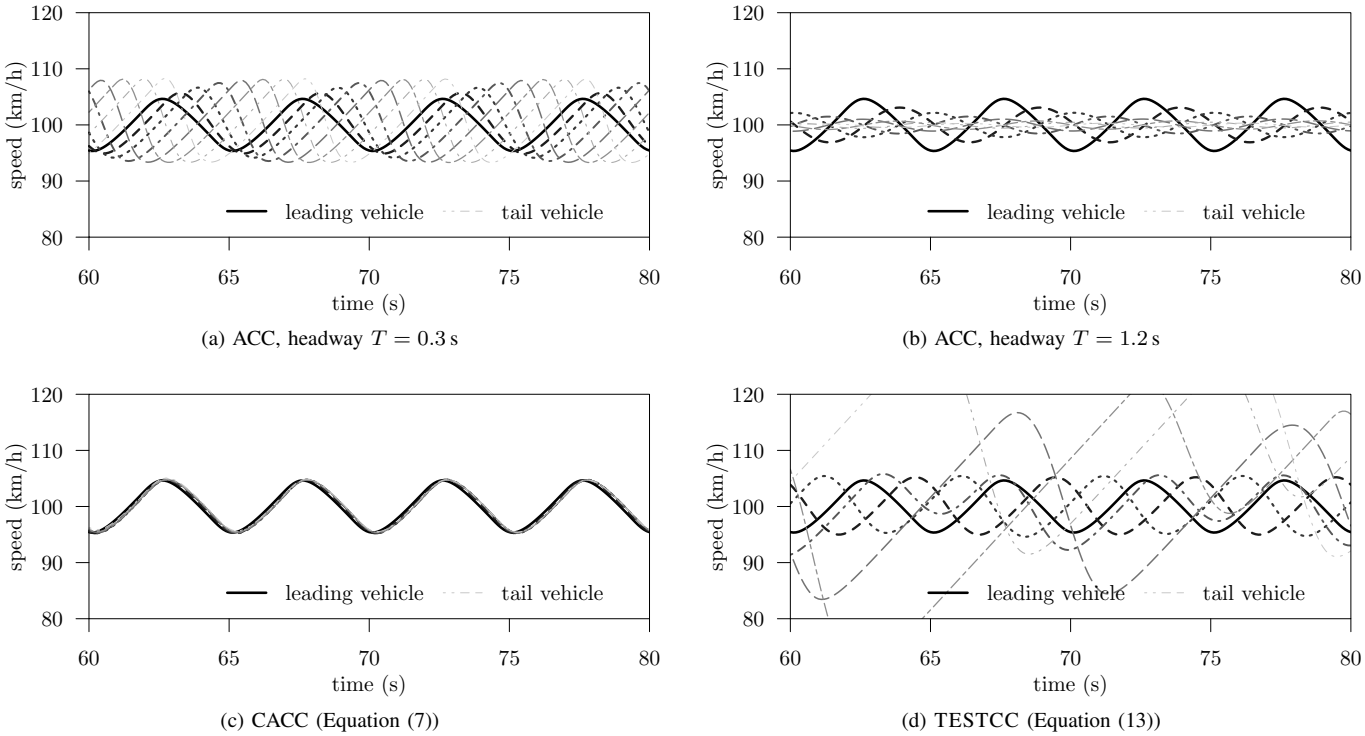
B. Join Maneuver

We use the same simulation parameters listed in Table I, but this time the leader drives with a constant speed, the platoon size is 4, and we test the scenario with the additional CACC parameters $\xi = 2$ and $\omega_n = 1$ Hz.

Figures 4 and 5 show the maneuver dynamics in terms of distance, speed, and acceleration, for two CACC settings ($\xi = 1,\ \omega_n = 0.2$ Hz and $\xi = 2,\ \omega_n = 1$ Hz, respectively). All figures describe in detail the maneuver in time. In the first 20 s, the platoon joins the simulations and the followers close their gap to the leader. Then, the joining vehicle enters the simulation with a speed of 100 km/h. After requesting to the leader the permission to join, it accelerates up to 130 km/h to reach the platoon. Once the joiner is roughly 15 m from the tail, it request the leader to conclude the maneuver and joins the platoon, speeding up a little to get closer to the last vehicle.

What it is interesting to notice is the change in dynamics between Figures 4 and 5. In Figure 5, the convergence is faster, as around 70 s the maneuver concludes, while in Figure 4 the maneuver ends roughly half a minute later. Faster convergence might however lead to instabilities or uncomfortable driving for the passengers. Accelerations and decelerations in Figure 5 are indeed stronger and even oscillate, thus requiring a proper parameter study to determine a good trade-off between convergence time and driving comfort.

This example shows the effectiveness of the simulator for implementing, simulating, and testing platooning maneuvers as well as control algorithms, helping researchers to analyze such systems in a realistic and trustworthy way.



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

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

(c) CACC (Equation (7))

(d) TESTCC (Equation (13))

Figure 3. Vehicles speed for the different implemented cruise controllers showing string-stability properties.

VII. Conclusion

We presented a new framework for the simulation of platooning systems called PLEXE, describing its structure and the longitudinal controllers it provides (i.e., CC/ACC and CACC). PLEXE is free to download, easy to customize, permits the simulation of mixed scenarios, and enables the realistic simulation of wireless networking and vehicle dynamics. The paper demonstrates the potential of the PLEXE simulator by considering two uses cases it can be used for, namely controller analysis and maneuvers implementation. We believe that PLEXE can be a valid research tool for the large-scale analysis and the comparison of automated car-following systems before their actual deployment.

References

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

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

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

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

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

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

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

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

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

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

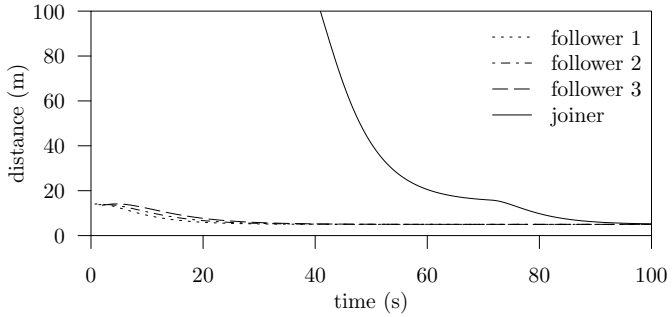
[11] P. Fernandes and U. Nunes, "Platooning of Autonomous Vehicles with Intervehicle Communications in SUMO Traffic Simulator," in *IEEE International Conference on Intelligent Transportation Systems (ITSC 2010)*, Madeira Island, Portugal, September 2010, pp. 1313–1318.

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

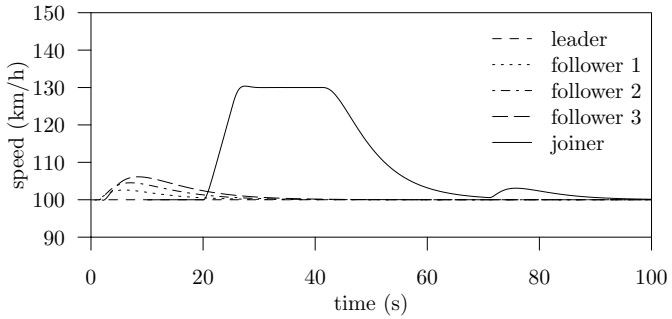
[13] A. Böhm, M. Jonsson, and E. Uhlemann, "Co-Existing Periodic Beaconing and Hazard Warnings in IEEE 802.11p-based Platooning Applications," in *10th ACM International Workshop on Vehicular Internetworking (VANET 2013)*. Taipei, Taiwan: ACM, June 2013, pp. 99–102.

[14] D. Jia, K. Lu, and J. Wang, "A Disturbance-Adaptive Design for VANET-Enabled Vehicle Platoon," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 527–539, February 2014.

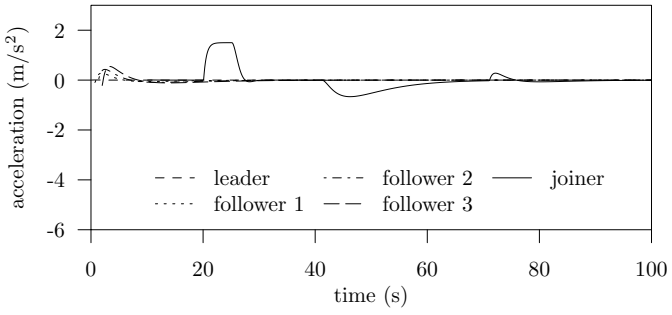
[15] L. Zhao and J. Sun, "Simulation Framework for Vehicle Platooning and Car-following Behaviors Under Connected-vehicle Environment," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 914–924, November 2013.



(a) distance

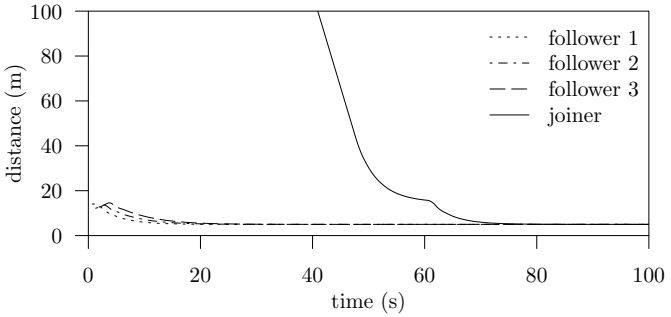


(b) speed

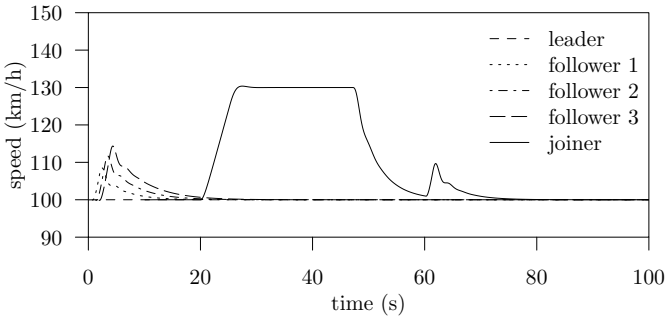


(c) acceleration

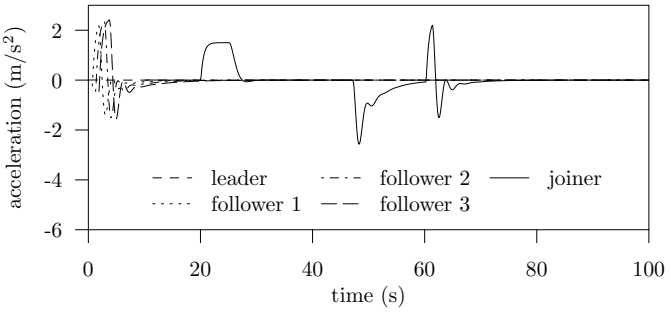
Figure 4. Vehicle dynamics for the join maneuver, $\xi = 1$, $\omega_n = 0.2\,\text{Hz}$.



(a) distance



(b) speed



(c) acceleration

Figure 5. Vehicle dynamics for the join maneuver, $\xi = 2$, $\omega_n = 1\,\text{Hz}$.

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

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

[18] J. I. Ge and G. Orosz, "Dynamics of Connected Vehicle Systems with Delayed Acceleration Feedback," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 46–64, September 2014.

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

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

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

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

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

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

[25] P. Fernandes, "Platooning of IVC-Enabled Autonomous Vehicles: Information and Positioning Management Algorithms, for High Traffic Capacity and Urban Mobility Improvement," PhD Thesis, University of Coimbra, Portugal, April 2013.

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

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

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