Computer Communications 212 (2023) 35–47

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# Co-simulated digital twin on the network edge: A vehicle platoon

Maurizio Palmieri [a,*], Christian Quadri [b], Adriano Fagiolini [c], Cinzia Bernardeschi [a]

[a] Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, Pisa, 56122, Italy
[b] Computer Science Department, University of Milan, Via Celoria 18, Milano, 20133, Italy
[c] Mobile & Intelligent Robots @ Panormous Laboratory (MIRPALab), Department of Engineering, University of Palermo, Viale delle Scienze, Building 10, Palermo, 90128, Italy

ARTICLE INFO

*Keywords:*
Digital twin
Co-simulation
Edge computing
Vehicle platoon

ABSTRACT

This paper presents an approach to create high-fidelity models suited for digital twin application of distributed multi-agent cyber–physical systems (CPSs) exploiting the combination of simulation units through co-simulation. This approach allows for managing the complexity of cyber–physical systems by decomposing them into multiple intertwined components tailored to specific domains. The native modular design simplifies the building, testing, prototyping, and extending CPSs compared to monolithic simulator approaches. A system of platoon of vehicles is used as a case study to show the advantages achieved with the proposed approach. Multiple components model the physical dynamics, the communication network and protocol, as well as different control software and external environmental situations. The model of the platooning system is used to compare the performance of Vehicle-to-Vehicle communication against a centralized multi-access edge computing paradigm. Moreover, exploiting the detailed model of vehicle dynamics, different road surface conditions are considered to evaluate the performance of the platooning system. Finally, taking advantage of the co-simulation approach, a solution to drive a platoon in critical road conditions has been proposed. The paper shows how co-simulation and design space exploration can be used for parameter calibration and the design of countermeasures to unsafe situations.

## 1. Introduction

During the last years, the development of technologies supporting autonomous vehicles has consistently increased. These technologies keep moving the field of autonomous vehicles toward regular deployment in our everyday life. The recurrent goal is to improve road traffic and long-distance transportation by optimizing several factors such as safety, costs, risks, and environmental sustainability by leveraging autonomous or semi-autonomous driving systems, like the *platooning* systems [1]. A platoon system relies on tight coordination among vehicles to achieve stability and short inter-vehicle distance. Data collected from onboard sensors are sent to other platoon members through a communication network.

Platoon-based vehicular systems are complex Cyber–Physical Systems (CPS) where a coordination algorithm interacts with the plant of the vehicles, consisting of the controller of the behavior of the vehicle and the physical elements of the vehicle. In this context, simulation techniques are suitable for handling the complexity of the CPS model in the development process, as they enhance the detection of any problems that may occur in the early stage of design before the development of the first prototype. The different subsystems of a CPS belonging to different domains need to be modeled with different languages and tools, each tailored to the requirements of the specific subsystem, so it is useful to exploit a co-simulation approach that enables the heterogeneous simulation using multiple tools [2].

A technology that is rapidly evolving and gaining the interest of many scientific research communities is the Digital Twin (DT), which consists of a virtual replica of physical processes created and maintained in order to gather insights about its physical counterpart [3]. A digital twin enhances the classical modeling approach because it involves the simulation of the model during the operational phase of the system; more precisely the physical element, during its operational life phase, continuously sends sensed data to its virtual counterpart, which combines the received data with the results of the simulation for monitoring, diagnostics, and predictive analysis and can autonomously provide commands to the physical elements to help it perform its assigned task, taking into account the results of the simulation.

The digital twin approach relies on multi-domain and multi-physics models, requiring the combination of different high-fidelity sub-models, e.g. networking model, thermal model, mechanical model, and electric model [4,5]. One of the many challenges identified by the research

* Corresponding author.
*E-mail address:* maurizio.palmieri@unipi.it (M. Palmieri).

https://doi.org/10.1016/j.comcom.2023.09.019
Received 1 February 2023; Received in revised form 26 June 2023; Accepted 16 September 2023
Available online 20 September 2023
0140-3664/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

community is the lack of a standardized approach for the creation and combination of the models of the physical world that compose the digital twin. Since these models belong to different domains, they require different tools and formalisms whose interoperability can only be achieved with great effort.

This paper exploits an approach of digital twin modeling based on the co-simulation technique to create multi-physics models of vehicle platoons suited for digital twin application, where, for example, tools that specifically address the modeling of the physical part of the system are coupled with tools modeling the information flow, using a well-known standard for co-simulation. The main contribution of the paper is to show how the modularity and flexibility of the approach can be used to easily set up various simulation scenarios and to derive an analysis of the system in different physical environments. In particular, in the case of safety-critical situations, in addition to simulation, the design space exploration feature, typically used for control parameter optimization, can be exploited for better calibration of parameters for safety.

The approach is introduced by means of a vehicle platoon example where the physical world is represented by a number of vehicles, their coordination algorithm, the road surface condition, and the network protocol and infrastructure, each modeled with specific tools. Results of the co-simulation runs show some of the advantages achievable with the proposed approach.

This paper is an extension of [6], and the newly introduced elements are: (i) a description of the proposed methodology for creating the models suited for a digital twin with a co-simulation approach, (ii) an application of the proposed approach with a more refined model of the vehicle, that takes into account more complex longitudinal dynamics of the vehicle, allowing the evaluation of the road surface on the performance of the platoon, (iii) a more detailed model of V2V communications that also takes into account the probability of packet delivery failure, (iv) an application of the cross-domain Design Space Exploration (DSE) where different co-simulation runs are executed with different combinations of parameters belonging to different domains, (v) the DSE is used to analyze the safety of the platoon under critical road surface conditions.

The remaining of this paper is organized as follows: Section 2 introduces related work on the main subjects of the paper: co-simulation and platooning. Section 3 introduces base information on the technologies involved in this paper, namely the Functional Mockup Interface co-simulation, the car platooning and the 5G and Multi-access Edge Computing network technologies for platooning. Section 4 describes the proposed approach, introducing the steps required for its application. Section 5 provides the different models used for the platoon co-simulation and their combination. Section 6 shows and discusses the obtained results. Section 7 highlights advantages and limits of the proposed approach. Finally, Section 8 concludes this paper, hinting at some future directions of this work.

## 2. Related work

Co-simulation (coupled simulation) allows the possibility to assemble a large simulation model out of pre-existing and possibly independently developed models, using the most appropriate simulators for each subsystem [7]. This is especially important for digital twin-oriented simulation of a CPS and for human-centered CPS modeling [8]. Among many others, recent examples of the application of co-simulation to cyber–physical systems are [9] where co-simulation is used to model a research vessel, suggesting that the co-simulation can be considered as an enabling technology for the digital twin of maritime vessels, [10] a technical report where a simple yet highly reproducible example of a digital twin is created and discussed, [11] where co-simulation and formal verification are used together to provide guarantees on the stability of the control algorithm for a brushless motor, and [12] where co-simulation has been successfully used for space coverage tasks of drones.

Platooning aims at improving road utilization and reducing fuel consumption thanks to the reduced inter-vehicle distance. Research communities have been working on platoons for almost four decades and count many successful research projects [13–17]. In [18–20] multiple experiments are conducted to quantify the fuel saving thanks to the drag reduction effect, showing a reduction in gas consumption up to 12%.

In order to achieve good performance in road utilization and fuel consumption the platoon formation must be preserved. In particular, inter-vehicle distance and string stability [21] properties must be guaranteed. To coordinate platoons vehicle-to-vehicle (V2V) communication technologies have been traditionally used, leveraging IEEE 802.11p and side link (PC5) 3GPP C-V2X [22,23]. Recently, an alternative option to the V2V paradigm has become available thanks to the introduction of a 5G network and edge computing, which can support latency-sensitive applications and computation close to the network edge. Some recent works [24–28] have proven the feasibility of edge-assisted platoon, moving the computation of the platoon instructions exploiting edge computing facilities. Moreover, this new paradigm overcomes the typical V2V problems, such as radio interference, shadowing, and multi-hop transmissions.

Simulating the motion of a vehicle requires the calculation of the traction force, generated on the wheels, and all applied resistance forces [29]. The traction force results from the complex interaction between the tire and the road and is a nonlinear function of driving motor torque and road types (see Burckhardt's [30]). All resistance forces include aerodynamic load losses and wind gusts (normally modeled as stochastic Dryden [31] winds). The driving torque can be obtained by various methods ensuring robustness and energy preserving [32–36].

As for the simulation framework for platooning (e.g., [24,37]), they are based on SUMO [38] (Simulation of Urban MObility), which provides only a partial model of the vehicle dynamics. For example, in [39] a model for a more realistic engine-braking system has been proposed and implemented in SUMO. However, this model neither considers the whole complexity of vehicle dynamics nor drag effects while our proposed approach not only includes road surface conditions but also wireless communications.

Recently, in [40] an integrated simulation tool for evaluating platoon safety is proposed, which includes a module for real-time prediction of communication delay. Adjustment of inter-vehicle gap according to safety constraints is suggested and Machine Learning modules are used as on-board predicting tools for communication, network and traffic parameters. The objective of the work is to analyze the impact of the transient communication outages on platoon safety. This work is based on ad hoc coupling of simulators: SUMO and OMNeT++.

Our proposed approach, instead, overcomes this limitation by exploiting standard interactions between models created from different tools thus allowing a co-simulation that combines network infrastructure with the detailed dynamics and control algorithms of vehicles. Moreover, design space exploration is suggested for investigating parameters setting for improving safety in critical scenarios. In [41], a common virtual test framework that incorporates multiple interacting and communicating vehicles is applied to a platooning application. Similarly to our work, model-in-the-loop simulation for a Cooperative Adaptive Cruise Control was conducted. With respect to [41], our approach is able to investigate and compare different communication technologies, showing that edge computing can be exploited to implement a mitigation strategy to improve the system safety.

## 3. Background

This section provides detailed background on the technologies involved in this work and on the platoon case study.

### 3.1. FMI co-simulation

The Functional Mockup Interface (FMI) [42] is a tool-independent standard for the co-simulation of dynamic systems. The main elements of an FMI-compliant co-simulation are the *Functional Mockup Units* (FMUs), which are responsible for simulating a single model in the specific formalism and execution environment used to create the model itself. An FMU may carry a whole simulation environment (tool wrapper FMU), or just information needed by an FMI-compliant host environment to simulate the model contained in the FMU (standalone FMU). FMU execution is orchestrated by a master algorithm, which is in charge of exchanging consistent data among the active FMUs. The master algorithm used in this work is Maestro2 [43], developed by the INTO-CPS Association, a result of the INTO-CPS project [44]. Orchestration is obtained through a set of standard APIs, defined by the FMI standard and implemented independently by each FMU, including, for example, initialization functions, functions for data exchange, and a function that moves the state of the FMU forward in time by one simulation step. At the time of writing, the FMI standard has just evolved to version 3.0, which extends the set of APIs allowing new features, such as, for example, improved event handling across FMUs and support for the packaging of virtual electronic control units into the FMUs [45].

The co-simulation approach based on FMI standard is different from other modular approaches, e.g. PLEXE [37] based on OMNeT++ simulator, and allows more flexibility: *(i)* different simulations components can be combined using FMI standard rather than relying on specific interfaces; *(ii)* each FMU can be developed in any programming language and *(iii)* the reuse of available simulation components is easier avoiding complex software rewriting.

### 3.2. Platooning

Platooning offers the opportunity to organize fleets of vehicles in groups traveling at short inter-vehicle distances reducing fuel consumption and enhancing road utilization. Maintaining platoon stability in dynamic and unpredictable road traffic conditions is challenging and requires tight cooperation among the platoon's vehicles. A recent and comprehensive overview of platoon coordination approaches is provided in [17]. In general, platoon coordination requires an intertwined system between onboard sensors/actuators and communication protocols.

A platoon control system is in charge of maintaining a specific inter-vehicle spacing policy (e.g., fixed space or constant headway time) and guaranteeing string stability irrespective of potential external perturbation. The control is realized through a control specifically designed and tuned to maintain the aforementioned platoon properties. One of the most popular control laws is Cooperative Adaptive Cruise Control (CACC) from the PATH project [15]. The goal of CACC control law is to provide specific longitudinal acceleration instructions to maintain fixed inter-vehicular distance and string stability. In particular, for each vehicle, CACC computes acceleration instructions using the data of the preceding and leader cars, in addition to the data of the vehicle itself. Data are exchanged among vehicles through wireless communication relying on network communication protocols, as it is discussed in the following.

Traditionally, the communication within a platoon is realized through V2V paradigm [22] using IEEE 802.11p and side link (PC5) 3GPP C-V2X [23]. This approach does not require any supporting infrastructure and the platoon is managed in a completely distributed fashion. In spite of the advantages in terms of deployment, V2V solutions suffer from several problems that limit their scalability and effectiveness. Uncoordinated access to radio channel is challenging because increases communication delays due to channel access contention. To overcome this limitation sophisticated transmission scheduling algorithms must be implemented onboard vehicles [46], however, these solutions do not completely solve the problem, in particular in scenarios of a high density of vehicles and high speed. Another drawback of V2V communication is the high sensitivity to radio shadowing phenomena specifically in presence of large vehicles, causing significant packet loss. Moreover, long platoons are not trivial to be managed due to the limited communication range of on-board radio which requires multi-hop communication incurring extra delays. To overcome these limitations a novel approach relying on the 5G mobile network and Edge Computing has been investigated as we will discuss in the next section.

### 3.3. 5G and MEC for platooning

The last generation of mobile network (5G) together with the introduction of Edge Computing architecture has unleashed new opportunities for the automotive domain. The 5G New Radio (5GNR) and the revolutionary concept of network slices allow network operators to deploy tailored services with stringent delay and reliability requirements. Moreover, the emergence of the Multi-access Edge Computing (MEC) standard architecture defined by ETSI [47] has provided a unified framework for deploying applications and services at the edge of the network [48]. MEC brings computation and storage capabilities to the edge of the network exploiting cloud computing virtualization technologies and dramatically reducing communication delays.

The automotive domain can benefit from these new technologies making possible the deployment of delay-sensitive supporting services. In this new setting, platooning can rely on the base stations to efficiently manage all radio communications through an effective MAC layer, by reducing inter-vehicle interference and shadowing effects. At the same time, the control of the platoon can be removed from vehicles and moved to a centralized service running on an edge server. This approach eases the support of long platoons and even multi-platoon compositions. Moreover, facilitates integration with other intelligent transportation systems already available at the edge.

In edge assisted platoon, each vehicle sends state information to the platoon controller which is implemented as a virtual network function deployed on an edge server. The platoon controller is responsible for computing acceleration instructions and sending them back to the platoon vehicles. All the communications between vehicles and platoon controller rely on the mobile network. Of course, an edge-assisted platoon depends on the availability of network coverage and suitable Quality of Service (QoS) level for guaranteeing effective message exchange and vehicle coordination. In particular, radio coverage represents an essential element of the system and edge-assisted platoon performance can be jeopardized by holes in coverage. However, the increasing deployment of mobile networks in metropolitan areas and along the main motorways diminishes the probability of a lack of coverage. More details about edge-assisted platooning are available in [24].

## 4. Proposed approach

This section describes the approach used for modeling, building, and prototyping digital twin for CPS. A real CPS is a complex system made of multiple intertwined components requiring expertise in different domains. FMI standard allows the decoupling design and implementation of distinct models focusing on their interaction interfaces.

Fig. 1 shows an example of CPS in the case of connected autonomous vehicles (CAV). We can observe four main components interacting with one another. The vehicle comprehends sensors and actuators, which are required for collecting the status and implementing the actions, as well as the physical dynamics of the vehicles (e.g., power train, braking and steering systems, wheels, etc...). The vehicle interacts with an external environment that models the road surface, weather conditions, obstacles, and everything outside the vehicle that could affect its motion. The coordination layer is responsible

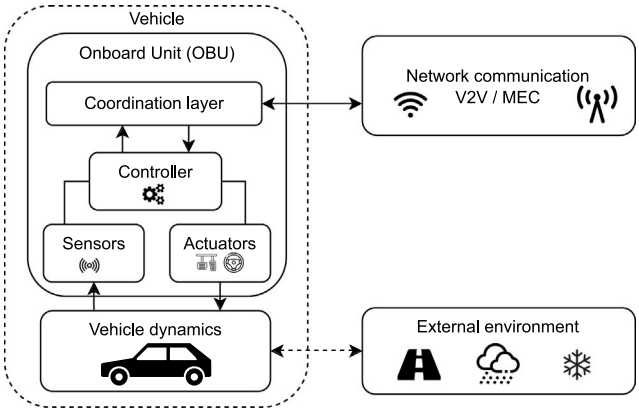



**Fig. 1.** Example of an agent.

for modeling the control law to generate the coordination instructions (e.g., longitudinal acceleration). Finally, the network component models the communication among vehicles, in the case of V2V platooning, or between vehicles and edge servers, in the edge-assisted platoon scenario (V2E platoon).

In the following, we present the multi-step methodology for modeling, building, and prototyping digital twin CPS. The first step consists of the bottom-up design of the CPS by modeling each (sub-)component independently exploiting specific expertise from different domains and the most suitable designing tool. Using the FMI standard we are able to define the interactions of the different modules highlighting the interplay among the components.

These interactions between the (sub-)components need to be included in the architecture of the FMI co-simulation, and they are represented by the connection between the FMUs. The initial co-simulation architecture can be used to validate the modeled system to make sure that the selected components fit correctly together and that the co-simulation runs exhibit the intended behavior in a non-critical, standard scenario (e.g. no obstacles, safe environmental conditions, and reliable network status).

Starting from the initial non-critical scenario, the next step introduces the safety requirements of the system (e.g., safety distance between vehicles and platoon stability) and exploits the co-simulation results to the identification of critical points that could lead to misbehavior and violation of the safety conditions (e.g., high network delay or prohibitive road conditions).

The following step is the configuration of testing scenarios focusing on challenging situations that stress the system leading to possible misbehaviors. Successively, the CPS model is co-simulated under the test scenarios analyzing the variation of behavior exploiting design space exploration tools. Thanks to the FMI-based design, this step is straightforward and can be easily automated using well-established tools [49].

The next step analyzes the results of co-simulated traces identifying the conditions where the safety requirements are violated. This process is performed both qualitatively, by means of GUI tools for checking the general behavior of the system, and quantitatively by measuring the level of requirement violation (e.g., the deviation from the target distance). This step provides a better understanding of the critical conditions that jeopardize the stability or even the safety of the system.

Following the analyses, it is possible to design and include mitigation techniques that eliminate or at least limit the most critical situations. In line with FMI-based design, they are implemented as (sub-)components directly in the CPS model by defining the parameters and the interactions among the other (sub-)components of the model. Then the whole methodology is iteratively repeated to assess the effectiveness of the mitigation techniques.

Summarizing, the proposed approach is based on:

- a multi-model paradigm, which allows different blocks to be modeled with different languages and tools;
- the FMI co-simulation standard, as a fundamental technology for the application of the DT approach;
- a toolchain with automated tools for co-simulation and design space exploration, as well as a graphical view of results;
- evaluation of critical/unsafe conditions and introduction of mitigation techniques;
- evaluation of the effectiveness of the proposed mitigation solutions.

In the next sections, we apply the described methodology to a specific use case of a cooperative platoon system.

## 5. Platoon modeling

This section explains the different model techniques used to create the different FMUs that compose the whole platooning system, and the last subsection shows how to combine these FMUs to study the co-simulation in the case of Edge architecture and in the case of V2V communications.

### 5.1. CACC FMU implementation

The mathematical formalization of the CACC [15] is described by the set of Eqs. (1), where the involved parameters have the following meaning:

- $C_1$ is the weighting factor between the acceleration of the leader and the acceleration of the preceding vehicle;
- $\xi$ is the damping ratio;
- $\omega_n$ is the controller bandwidth;
- $d_{des}$ is the desired distance among the vehicles;
- $l_{i-1}$ is the length of the vehicle;

and the variables used are:

- $\boldsymbol{x_i}$ is the position of the $i$th vehicle, $\dot{\boldsymbol{x}}_{\boldsymbol{i}}$ is its speed and $\ddot{\boldsymbol{x}}_{\boldsymbol{i}}$ is the acceleration; in particular $\dot{\boldsymbol{x}}_{\boldsymbol{0}}$ is the speed of the platoon leader, and $\ddot{\boldsymbol{x}}_{\boldsymbol{0}}$ is its acceleration;
- $\boldsymbol{x_{i-1}}$ is the position of the preceding car (w.r.t. the $i$th car), $\dot{\boldsymbol{x}}_{\boldsymbol{i-1}}$ is its speed and $\ddot{\boldsymbol{x}}_{\boldsymbol{i-1}}$ is the acceleration;
- $\boldsymbol{\varepsilon_i}$ is the distance error between the vehicle $i$ and the preceding vehicle w.r.t. the target distance $d_{des}$;
- $\dot{\boldsymbol{\varepsilon}}_{\boldsymbol{i}}$ is the delta speed between the $i$th vehicle and the preceding one.

$$
\begin{aligned}
\dot{\varepsilon}_i &= \dot{x}_i - \dot{x}_{i-1} \\
\varepsilon_i &= x_i - x_{i-1} + l_{i-1} + d_{des} \\
\alpha_1 &= 1 - C_1, \alpha_2 = C_1 \\
\alpha_3 &= -\left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)\omega_n \\
\alpha_4 &= -C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\omega_n \\
\alpha_5 &= -\omega_n^2 \\
\ddot{x}_{i_des} &= \alpha_1\ddot{x}_{i-1} + \alpha_2\ddot{x}_0 + \alpha_3\dot{\varepsilon}_i + \alpha_4(\dot{x}_i - \dot{x}_0) + \alpha_5\varepsilon_i
\end{aligned}
\tag{1}
$$

These equations have been easily expressed in C code and then the FMU has been generated with PVSio-web [50]. The resulting FMU has 7 input variables ($\boldsymbol{x_i}$, $\dot{\boldsymbol{x}}_{\boldsymbol{i}}$, $\boldsymbol{x_{i-1}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{i-1}}$, $\ddot{\boldsymbol{x}}_{\boldsymbol{i-1}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{0}}$, $\ddot{\boldsymbol{x}}_{\boldsymbol{0}}$) and produces 1 output ($\ddot{\boldsymbol{x}}_{\boldsymbol{i_des}}$) using the standard parameters values shown in Table 1.

**Table 1**
CACC parameters values.

| Parameter | Value | |
|---|---|---|
| CACC parameters | | |
| $C_1$ | 0.5 | |
| $\xi$ | 1 | |
| $\omega_n$ | 0.2 | |
| $l_{i-1}$ | 4 | m |
| $d_{des}$ | 10 | m |

### 5.2. Platoon FMU implementation

The MATLAB/Simulink suite [51] is one of the most renowned and qualified environments supporting block-based modeling. Simulink bridges development from requirements and system architecture thanks to a rich library of small components (e.g. PID controllers, integrator blocks) easily combined through a flexible user interface based on the drag-and-drop approach.

Each vehicle of the platoon is modeled in Simulink with a couple of sub-models:

- a kinematic model capturing the speed constraints of the car;
- a dynamics model that produces the acceleration of the car, taking into account environmental factors such as wind and road surface friction.

The kinematic model takes as input the acceleration $a$ of the car and updates speed $v$ and position $x$ as shown in Eqs. (2).

$$\begin{aligned} \dot{v} &= a \\ \dot{x} &= v \end{aligned} \tag{2}$$

The equations adopted in this model represent a first-order system, implemented using two Simulink standard integrator blocks, whose initial values (initial speed and position) are stored in two model parameters.

The dynamic model, inspired by [35] and shown in Fig. 2, takes as input the desired acceleration $\ddot{x}_{des}$, the desired speed $\dot{x}_{des}$, and actual speed $v$, and it produces the actual acceleration $\ddot{x}$ of the car according to Eqs. (3), where $F_a(v)$ is the aerodynamic drag, $\mu$ is the road friction, $T$ is the traction force, $F_z$ is the total normal load of the car, $\sigma$ is the slip ratio, $\omega$ is the wheel rotational speed.

$$\begin{aligned} F_a(v) &= \frac{1}{2}\rho C_x S v^2 \\ \mu &= c_1(1 - e^{-c_2\sigma}) - c_3\sigma \\ T &= mR(\ddot{x}_{des} + k(v - \dot{x}_{des})) + R(F_a(v) + W) \\ F_z &= \frac{mg}{2} + \frac{h}{l}\ddot{x}_{des} \\ \sigma &= \frac{\omega R - v}{\omega R} \\ \dot{\omega} &= \frac{1}{I_\omega}(T - R\mu F_z) \\ \ddot{x} &= \frac{1}{m}(\mu F_z - F_a(v) - W) \end{aligned} \tag{3}$$

In particular, the computation of the actual acceleration $\ddot{x}$ depends on the mass of the car $m$, the height $h$ of its center of mass from the ground, the aerodynamic drag coefficient $C_x$, its frontal area $S$, the wheel radius $R$, the air density $\rho$, the gravity acceleration $g$, the wheelbase $l$, and a control gain $k$ specifying the convergence rate of the tracking speed error, $v - \dot{x}_{des}$. Moreover, the traction signal is computed based on Burckhardt's Road Surface Friction model (RSF) [30], in which $c_1$, $c_2$, and $c_3$ are coefficients depending on the type of surface (i.e. wet, snowy, icy). Finally, the wind disturbance signal $W$ is generated according to Dryden's model which is a stochastic signal obtained from a Gaussian white noise that is filtered by a suitable transfer function as in [31].

As numerical values for the above parameters, we chose the nominal values reported in [35].

It is possible to create a single Simulink model where any number of vehicles is modeled by replicating the two sub-models many times; by doing so, it is possible to easily obtain a model of the platoon.

As far as it concerns the platoon's leading car, it is possible to use a model whose acceleration is provided by a Simulink source block representing a driver's behavior (e.g. a sequence of trapezoidal signals simulating a series of acceleration and braking commands, or a sinusoidal wave for smooth changes). Such a platoon model is subject to a constraint on the initial position of the vehicle: the cars should be positioned according to their order in the platoon and the leading car must be positioned in front of the platoon; in other words, users should pay attention to the parameter value used for the initial position of the integrator block of the leader's kinematic sub-model.

Fig. 3 shows the model for the leading vehicle. It is possible to see that the desired acceleration of the leader is provided through a signal builder block where it is possible to design custom signals to match different operations of the vehicle. In this specific model, the chosen signal is a train of trapezoids with a height of 0.5 $m/s^2$. The inputs of the dynamics sub-model are the desired acceleration, the desired speed (which is the integral of the desired acceleration), and the actual speed. The output of the dynamics sub-model is the actual acceleration, which is forwarded to the kinematics block in order to produce actual speed and actual position. The actual speed is then fed back to the dynamics sub-model.

The model of the other vehicles is equivalent to the one of the leader, with the only difference being that the desired acceleration is provided as an external input. As a final remark on the model, Simulink provides a native feature to export the whole model as a single FMU, compliant with FMI standard, with the same number of inputs, parameters, and outputs as the model.

Fig. 4 shows the model for a 5-vehicle platoon, where 4 cars (violet background) follow a leading one (green background). The model has 4 inputs values (the 4 desired accelerations of the succeeding cars) and produces 15 outputs variables (acceleration, speed, and position of all the vehicles) along with 4 more outputs that represent the distance between all couples of subsequent cars in the platoon.

### 5.3. V2V FMU implementation

In order to introduce network communication in the V2V architecture, a Simulink model has been created. This model takes as input three values (acceleration, speed, and position) for each vehicle and merges them into a single message with the typical V2V frequency of 10 Hz. For each message, the model computes a random delay that represents the MAC access delay (e.g. block `TX Leader` in Fig. 5), using an exponential distribution with a mean value of 20 ms as suggested in [52]. Then, for each intended destination of the message, we used the model implemented in [53] to quantify the Packet Delivery Ratio (PDR) to randomly assess if the packet is successfully received (e.g. blocks `RX Leader` in Fig. 5). The PDR is based on the actual distance between the source of the message and the destination.

Using such a model, it is possible to export an FMU from Simulink that takes as input 15 variables (acceleration, speed, and position of the leader and of the four following vehicles) and produces as output the acceleration, speed, and position of the leader received by each following vehicle (12 output variables), and, for each following vehicle, the acceleration, speed, and position of the preceding vehicle (9 outputs variables). It is important to notice that from the point of view of the first following vehicle, the leader and the preceding one are the same. Fig. 5 shows the detail of the FMU; for the sake of simplicity, a simple 3-vehicles platoon is shown. Moreover, the figure uses a compact notation where *xX*, *vX*, and *aX* stand for, respectively, the position, speed, and acceleration of vehicle X in the platoon, considering vehicle number 1 as the leader.

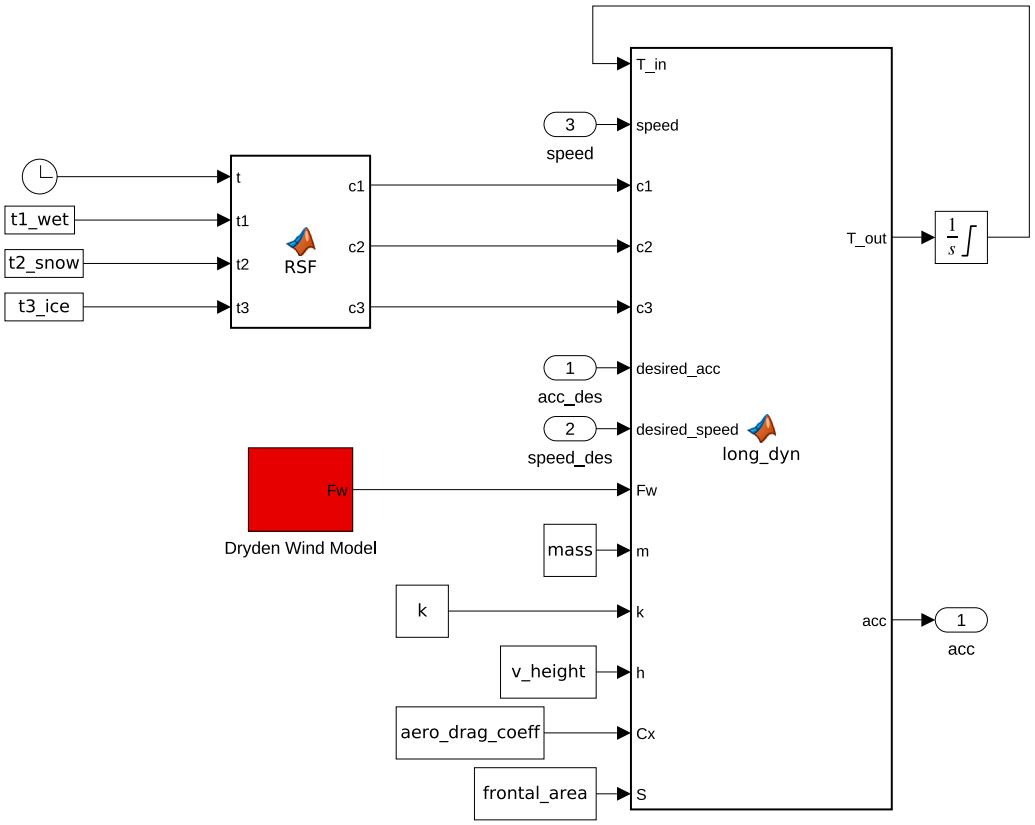



**Fig. 2.** Dynamics model of the vehicle.

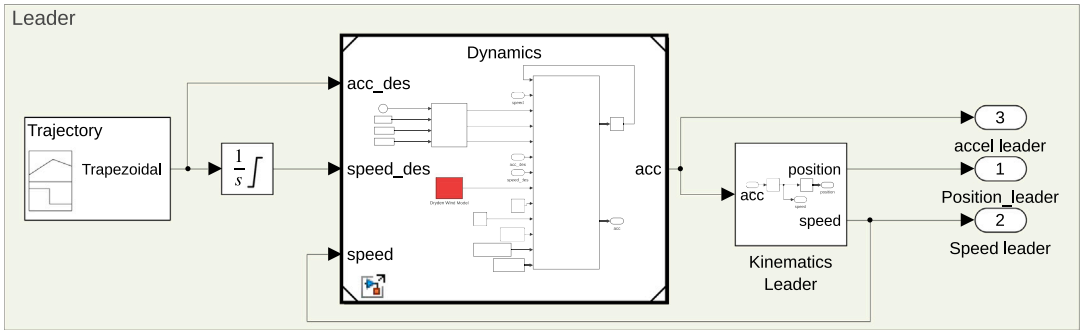



**Fig. 3.** Vehicle model.

### 5.4. Edge network FMU implementation

The edge network is modeled by extending the standalone discrete simulator presented in [24] to support FMI standard using UniFMU [54]. This module simulates the whole mobile-edge network including the user equipment (UE) and is written in Python programming language. The module is made of three sub-modules: *(i)* the network application running on-board each vehicle, which periodically reads the data from the on-board sensors and sends them to the network; *(ii)* the radio access network (RAN) which simulates the base stations and manages the handover; and *(iii)* the edge computing facility which simulates computational resources at the edge and the platoon controller edge application. In Fig. 6 we show the schema of the edge network module. The black arrows represent the uplink data flow carrying all the information of onboard sensors, while the red arrows refer to the downlink flow for delivering the desired acceleration. The platoon controller, in the edge sub-module, is in charge of collecting all data coming from vehicles and identifying which vehicles depend on the received data. In this work, we employ the *leader-and predecessor-following* control topology which is the most suitable for CACC control law.

Managing the platoon through the edge is subject to higher communication delays than traditional V2V architecture[1] because more network nodes are involved in the communication between vehicles and the edge controller. Moreover, mobile network delays are not symmetric in general, usually, uplink delays are greater than downlink ones. We model this characteristic using a pair of independent random variables configurable through FMU parameters. Besides, we allow the user to model the network delays at different levels of granularity. by specifying uplink and downlink delays as simple random variables or a chain of independent random variables. The second option allows for

[1] This is true only when radio channel contention is low, in presence of high vehicle density communication delays may increase due to higher contention and interference.

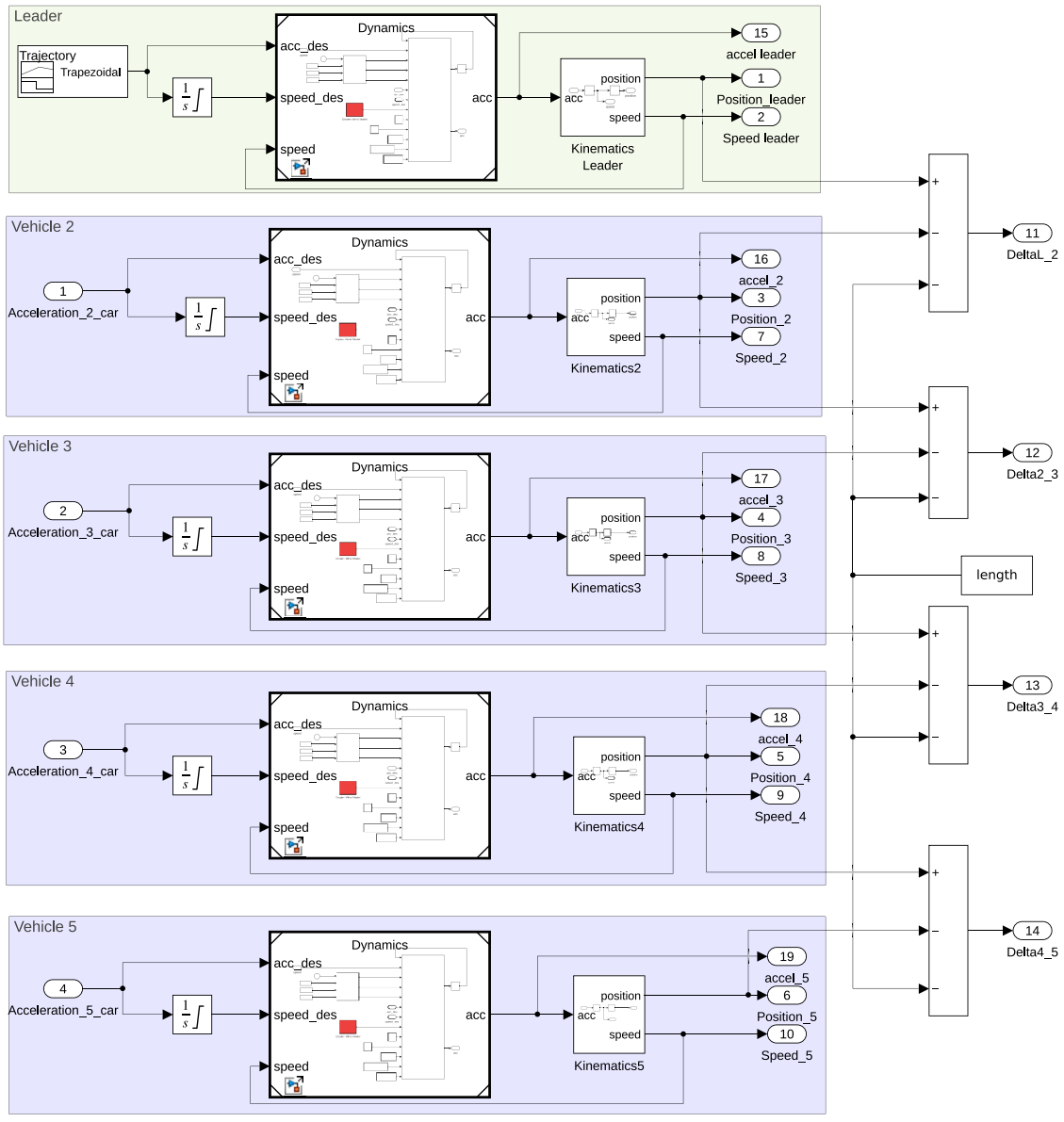



**Fig. 4.** Simulink Platoon model.

an arbitrarily complex model of mobile network as the one presented in [55].

### 5.5. Co-simulation architecture

Fig. 7 shows the comparison between the two adopted architectures for the 3-vehicles platoon. In both cases, $N+2$ FMUs are involved:

- the one with the physical laws defining the platoon, explained in Section 5.2. The choice of having all the vehicles in a single FMU simplifies the execution of the co-simulation because it requires only one active instance of MATLAB to execute;
- the one with the network communication model (either the MEC one generated from UniFMU, explained in Section 5.4 or the one with the V2V channel explained in Section 5.3);
- $N$ FMUs for the vehicle controller generated with PVSio-web, explained in Section 5.1, where $N$ is the number of following vehicles ($N = 2$ for the 3-vehicles platoon).

The main difference between the two architectures is the transmission of the desired accelerations evaluated by the CACC algorithm. In the MEC case, the values are transmitted to the corresponding vehicle through the MEC infrastructure and so the outputs of CACC FMUs are connected to the MEC FMU which is in charge of transmitting them to the corresponding vehicle in the Platoon FMU. On the other end, in the V2V case, the CACC is computed on-board of the vehicle and therefore the output of the CACC FMU is directly connected to the corresponding vehicle in the Platoon FMU.

## 6. Co-simulation results

In this section we execute the co-simulation of a 5-vehicles platoon. The co-simulation architectures introduced above are simulated choosing different deployment scenarios with respect to the road surface and observing the behavior of the platoon. Moreover, the DSE tool is applied to analyze the system for different values of platoon parameters.

All the co-simulations have been executed with a fixed-step co-simulation algorithm with a step size of 10 ms. This value has been chosen because it matches the order of magnitude of the delays taken into account in the considered cases; it is possible to use a lower step

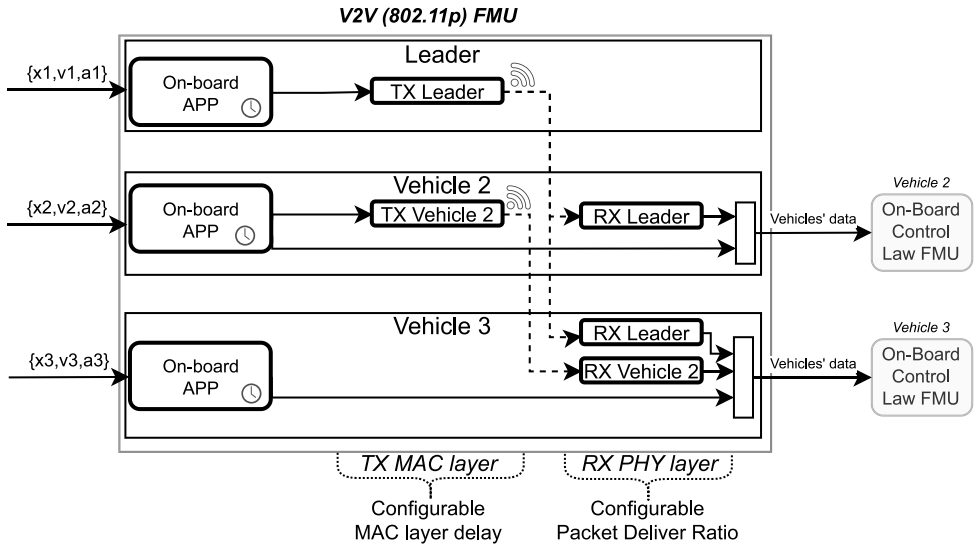



**Fig. 5.** V2V (802.11p) FMU model.

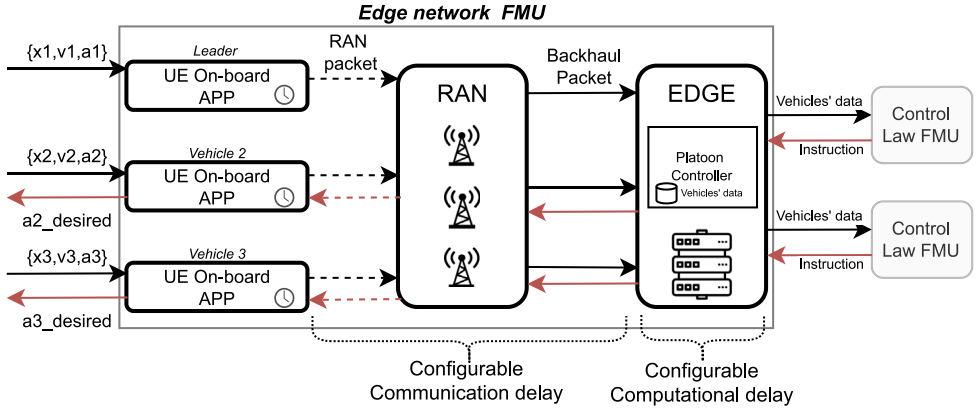
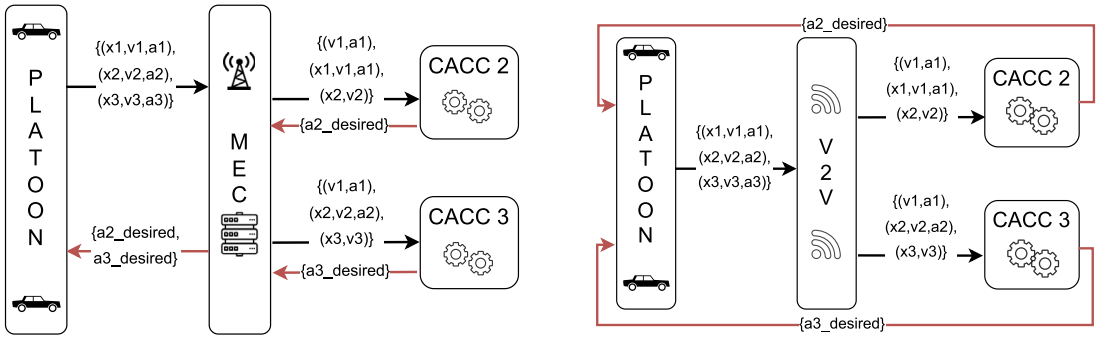



**Fig. 6.** Edge network FMU model.




(a) MEC co-simulation case.

(b) V2V co-simulation case.

**Fig. 7.** Comparison of co-simulation cases for the 3-vehicles platoon.

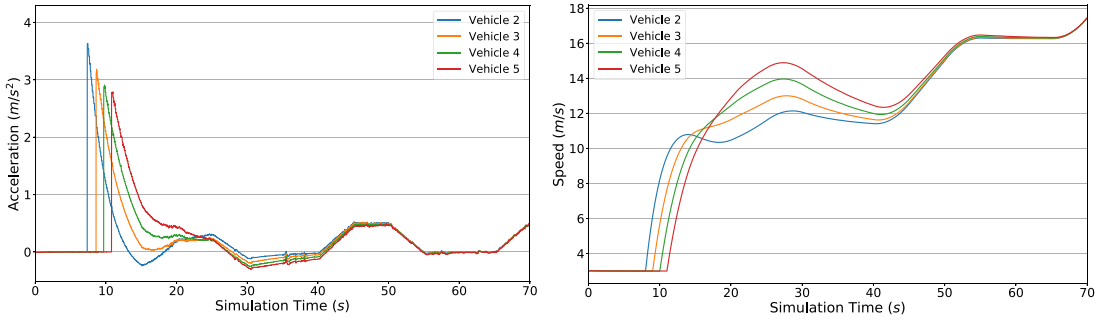
size but it will lead to an increase in the required time to run the co-simulation; a lower step size will give similar results, at the cost of more time required to run the co-simulation. All the co-simulations have an end time of 70 s. As far as concerns the parameters for the different communication schemes, Table 2 summarizes the chosen values.

In Section 6.1, a dry road surface is considered; while in Section 6.2 the road surface changes from dry to icy, causing some problems to the platoon that can be investigated with our approach.

### 6.1. Non-critical scenario

In this scenario, we have considered a standard and safe situation where the road surface is always dry. The main purpose of this scenario is to validate the models and compare the behaviors of the two different communication schemes (V2V and MEC).

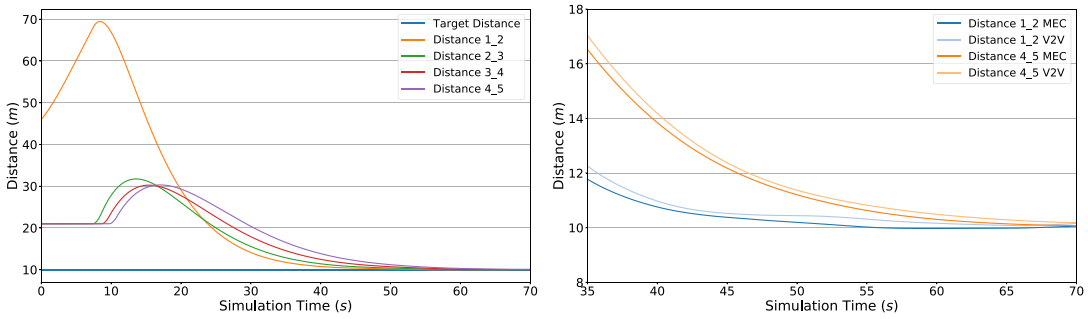
From a macroscopic point of view, the platoon shows the same behavior for both communication schemes. The results of simulations




(a) Acceleration evolution through the whole simulation.

(b) Speed evolution through the whole simulation.

**Fig. 8.** Co-simulation with a dry surface for both V2V and MEC.




(a) Distance results for both V2V and MEC.

(b) Cases comparison between two couples of vehicles.

**Fig. 9.** Comparison of distances with a dry surface.

**Table 2**
V2V and MEC parameters values.

| Parameter | Value | |
|---|---|---|
| V2V parameters | | |
| Packet Size | 190 | Bytes |
| Transmission power | 23 | dBm |
| Data rate | 6 | Mbps |
| Vehicles density | 120 | veh/km |
| Exponential average delay | 20 | ms |
| MEC parameters | | |
| Fixed delay UL | 10 | ms |
| Exponential average delay UL | 10 | ms |
| Fixed delay DL | 10 | ms |
| Exponential average delay DL | 5 | ms |

can be summarized in Fig. 8, which shows the accelerations and the speeds of the vehicles of the platoon, except the leading car.

More precisely, in Fig. 8(a) it is possible to notice that at the beginning of the co-simulation all the vehicles accelerate to reach the target distance with the preceding vehicle and then, starting from around time 50 all the vehicles converge to the same acceleration, the train of trapezoids mentioned in Section 5.2. and the platoon reaches the stability.

Fig. 8(b) represents the behavior of the speeds of all the cars. As expected, the speed is the integral of the acceleration and therefore all the speeds converge to the same value of approximately 17 meters per second at the end of the simulation.

Since the two communication schemes produce similar speeds, also the distances among the vehicles are macroscopically similar. In Fig. 9(a) it is possible to see that all the distances converge to the target distance of 10 meters in roughly 50 s. However, the behavior of the platoon is not exactly the same. As an example, Fig. 9(b) shows a detailed comparison between the MEC and V2V schemes considering the distance between the first two vehicles (`Distance 1_2` in the legend) and the distance between the last two vehicles of the platoon (`Distance 4_5` in the legend). The distance between the first two cars converges with the target distance earlier than the distance between the last two cars. Moreover, the distances in case of the MEC infrastructure converge always faster than the distances in case of the V2V one.

*Design space exploration analysis*. In the following we show an example of application of the DSE tool integrated in the co-simulation framework for computing the mean error between the actual and the target distance between vehicles in the platoon, assuming different combination of values of parameters target distance and initial speed of vehicles.

Table 3 shows the values of the mean error, assuming different combination of values for the parameters target distance (ranging over 20 m, 15 m, 10 m) and initial speed (ranging over 3 m/s, 5 m/s) of vehicle 2 and vehicle 4. Initial speed of the other vehicles is fixed at 3 m/s. In the table, **Distance** is the target distance; **Speed 2** and **Speed 4** are the initial speed of vehicle 2 and vehicle 4; **Error MEC** and **Error V2V** are the mean error computed by the analysis in the two communication schemes.

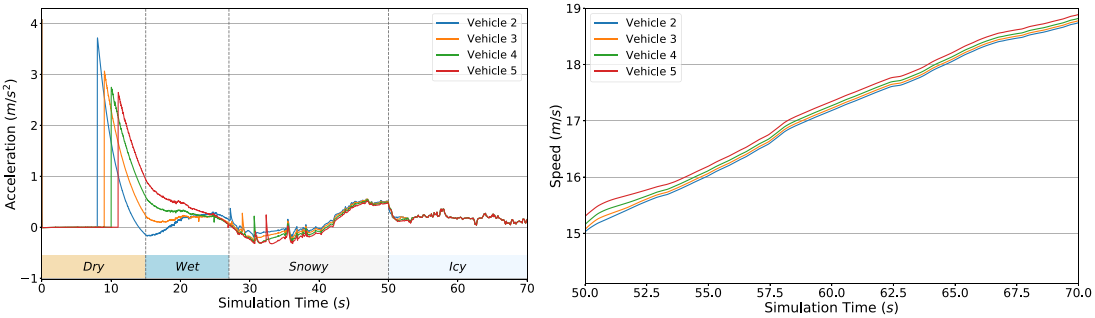
The mean error is evaluated as follows: (i) for each couple of subsequent vehicles, evaluate the absolute value of the difference between their distance and the target distance, (ii) sum the 4 differences and then (iii) average the sum over time.

To avoid the initial transient where the error is high, the mean error is only evaluated in the second half of the co-simulation. By comparing the two communication schemes it is possible to state that the mean error of the MEC infrastructure is always lesser than the one of the V2V, for the considered combinations of parameters.

Overall, the application of the proposed approach showed that the MEC infrastructure exhibits more stable behavior and thus less error than the V2V case.

### 6.2. Critical scenario

In this deployment scenario, we have considered a dangerous situation where the road surface changes quickly from dry to icy. More




(a) Acceleration evolution through time.

(b) Speed evolution in the last 20 seconds.

**Fig. 10.** Co-simulation with the icy surface for both V2V and MEC.

**Table 3**
DSE analysis for MEC and V2V.

| Distance (m) | Speed 2 (m/s) | Speed 4 (m/s) | Error MEC (m) | Error V2V (m) |
|---|---|---|---|---|
| 20 | 5 | 5 | 1.26 | 2.82 |
| 20 | 5 | 3 | 1.32 | 3.06 |
| 20 | 3 | 5 | 1.40 | 3.17 |
| 20 | 3 | 3 | 1.53 | 3.40 |
| 15 | 5 | 5 | 1.44 | 3.26 |
| 15 | 5 | 3 | 1.53 | 3.50 |
| 15 | 3 | 5 | 1.60 | 3.61 |
| 15 | 3 | 3 | 1.70 | 3.84 |
| 10 | 5 | 5 | 1.64 | 3.71 |
| 10 | 5 | 3 | 1.75 | 3.95 |
| 10 | 3 | 5 | 1.89 | 4.05 |
| 10 | 3 | 3 | 2.01 | 4.29 |

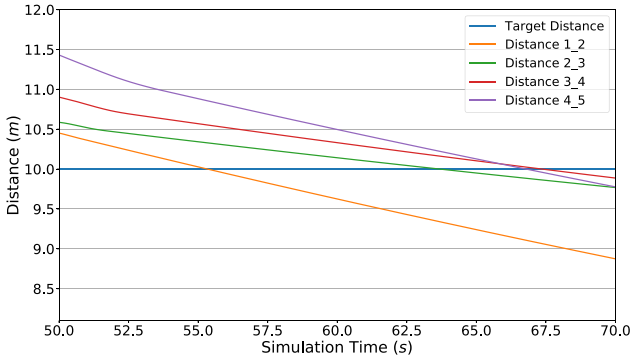



**Fig. 11.** Distances evolution with the icy surface.

precisely, the road surface is dry for roughly the first 15 s, then it is wet until around time 27, then it is snowy till around 50, and then gets icy till the very end. This scenario is obtained by simply changing three parameters in the platoon FMU that represent the time at which the road surface changes from dry to wet (15 s), from wet to snowy (27 s), and from snowy to icy (50 s). All the other parameters are exactly the same used in the previous, non-critical, scenario.

Also in the case of different road surfaces, from a macroscopic point of view, the platoon shows the same behavior for both communication schemes. Fig. 10 shows the accelerations and speeds of the following vehicles. In Fig. 10(a), it is possible to notice that, when the road surface is wet the behavior of the vehicles is very close to the one of the non-critical scenario (see Fig. 8(a)).

The rapid transition to a snowy surface only leads to a peak in the acceleration of the cars but in the last 20 s (i.e. when the road becomes icy), the acceleration change drastically, and all the vehicles converge to the same behavior different from the desired trapezoidal train.

**Table 4**
DSE analysis of the changing time of the target distance.

| Safe | Changing time (s) |
|---|---|
| True | 47 |
| True | 48 |
| False | 49 |
| False | 50 |

Fig. 10(b) shows the behavior of the speeds in the last 20 s of the simulation, and it is possible to see that all the vehicles are faster than their preceding ones.

This is the worst result ever as it means that the cars will eventually collide, as confirmed by Fig. 11, which shows the evolution of distances through the last seconds of the simulation and clearly shows that, in the end, these distances are below the target distance (the blue line in the figure) and tending to zero.

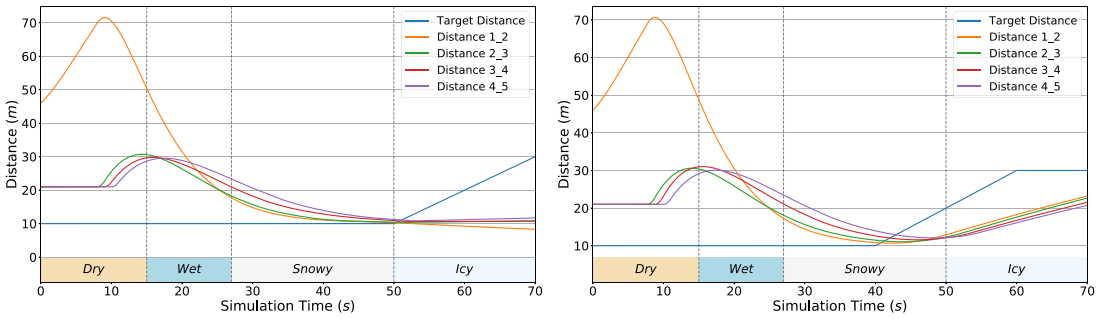
### 6.2.1. Mitigation

Possible mitigation to this problem is the usage of a dynamic target distance. Assume the vehicles linearly increase the target distance from 10 to 30 m, as soon as they perceive the icy road. Dynamically varying the target distance leads to a temporary loss of string stability property, however, in critical conditions, preventing vehicle collisions is a priority w.r.t. the platooning properties.

*V2V communications*. In case of V2V communications icy road is perceived exactly at time 50, when the road changes from snowy to icy. As shown in Fig. 12(a), for the V2V case, this strategy is not sufficient, as the distance between vehicles 1 and 2 still tends to zero.

*MEC communications*. As far as concerns the MEC infrastructure, we assume that the platoon controller can benefit from integration with other MEC services in charge of monitoring and predicting road surface, and can promptly trigger the mitigation maneuvers in advance with respect to the V2V case.

With the DSE analysis, it is possible to search for different values of reaction time, i.e. the time at which the vehicles start increasing the target distance, and find a value that avoids the collision. Table 4 shows the results of such analysis, where **Safe** equal to “false” means that there is at least one distance between any 2 subsequent vehicles that drops below 10 meters (leading to a dangerous situation and possible collision), “true” means that no distance drops below 10 meters (leading to a safe condition) and **Changing Time** is the time at which the cars start increasing the target distance. It is possible to state that reacting 2 s earlier with respect to the V2V case (50 is the time at which icy road is physically perceived) is sufficient to avoid the collision, as shown in Fig. 12(b) where the **Changing Time** is 40 and it is possible to see that all the distances do not fall below 10 m.




(a) V2V results (target distance changing time 50).

(b) MEC results (target distance changing time 40).

**Fig. 12.** Comparison of distances with the target distance in case of the proposed mitigation.

## 7. Discussion

The co-simulation approach to CPS modeling and analysis proposed in this paper essentially consists in modeling the controller, the physical parts and the communication network with domain-specific tools. The main advantages of this approach are modularity and flexibility of the co-simulation architecture. In addition, by exploiting the FMI standard, models at different levels of abstraction can be interchangeably used for a specific component, only replacing the FMU of the component, without affecting the other components. For example, the V2V communication channel was first modeled by a delay FMU (see the conference paper), and successively refined with a more detailed model in this paper.

Another benefit of the proposed approach is the component reusability. As discussed in Section 3.1, co-simulation based on FMI standard makes the usage of previously developed components easier than standalone/monolithic simulation approaches. As a matter of fact, we combined different components, some of which have been previously developed by the authors (e.g. MEC [24] and Dynamics/Kinematics [35] of a Vehicle), while others have been taken from the literature (e.g., V2V communication module [53]). Moreover, the re-usage of already developed components allows fast prototyping simulation environments for digital twins.

Simulation of a real-world environment is computationally expensive. In particular, the size and the level of realism of the model strongly affect the simulation time and the amount of required computing resources. For these reasons, the modeling and the design of the co-simulation should balance the model complexity and computation time according to the goal of the experiments. For example, the number of vehicles in the platoon is a parameter that strongly influences the simulation time. In our experiment, we simulate a platoon of 5 vehicles, which represents a trade-off between tractable and significant scenarios. To partially overcome this limitation, techniques like distributed co-simulations could be used to improve the framework's scalability, allowing the modeling and simulation of larger and more complex systems.

In addition, many different real-world environment factors can be modeled in the digital prototype, for example, road surface, drag effect, wireless medium, and even driver behavior. All these aspects can be modeled with domain-specific tools and can be put in co-simulation (e.g., Simulink, OpenModelica, C-language, ···). Obviously, to keep the model and the analysis tractable, a careful selection of the most relevant features is essential, taking into account both the objective of the analysis and the resulting model complexity.

In the same vein, the prototype can also be used to evaluate the robustness of the distributed controller with respect to vehicle model uncertainties. This includes, for example, the possibility of testing and evaluating the impact of imprecise knowledge of vehicle parameters, such as mass, drag coefficient and its frontal area, on the overall performance of the system; in fact, the values of all these parameters are known only nominally, and so testing the effectiveness of the controllers used, based on these values, becomes an essential feature. Of course, establishing the amount of parameter uncertainty that does not cause instability is a challenge and an open problem, but the co-simulation approach can be used to provide some conservative limits and boundaries.

The proposed co-simulation approach introduces a non-negligible computational overhead due to the co-simulation architecture of the FMI standard. Indirect interaction among modules leads to longer simulation time compared to standalone/monolithic simulators. However, this limitation is largely compensated by the high degree of flexibility offered by the co-simulation.

Finally, the proposed framework natively supports the design space exploration analysis, that can be applied for calibration of system parameters with respect to an objective function. This technique has been successfully used to validate a mitigation strategy for a safety issue, as shown in Section 6.2.1.

## 8. Conclusions

In this paper, a co-simulation based approach for building models that can compose the digital twin of a platoon-based Vehicular CPS has been presented. During the co-simulation, the platoon coordination algorithm is coupled with a model for vehicle dynamics and a module that includes a network communication protocol for the exchange of data among the vehicles.

The FMI co-simulation standard has been used to integrate the physical component with the Edge-based infrastructure and to analyze and evaluate the results. The heterogeneity of the different sub-models has been managed by different tools for the FMU creation, *PVSio-Web* for the C code implementing the CACC algorithm, *Simulink* for the differential equations describing the plant behavior and *UniFMU* for the Python module of the network simulation.

Moreover, we have shown the benefits of the co-simulation approach, which has allowed us to easily prototype different scenarios by exploiting the FMI standard and reusing already available models. In particular, the results show that in the non-critical scenario, the CACC algorithm performs in a correct manner, satisfying stability and communicating the desired acceleration to the physical system. However, the safety of the system is not guaranteed in the critical scenario where the road surface condition changes quickly into an icy road.

Following the steps of the proposed approach, the co-simulation has provided data on a possible mitigation technique consisting of increasing the inter-vehicle target distance when the road becomes icy. The co-simulation experiments show that knowing the information about the road condition in advance can improve the safety of the platoon vehicles.

Starting from the work presented here, possible future developments are:

- **Extend the proposed approach with guidelines on the actual implementation of the digital twin**, including the communication between the physical world and the digital twin, exploiting, for example, the MQTT protocol, as suggested in [56],
- **Enhancing the network modeling, including channel behavior**, for example using OMNeT++ [57] to model V2V or MEC communications.
- **Explore distributed co-simulation** as a technique for enhancing the prototyping of large CPS, splitting the computational demand on different machines.
- **Studying threats related to cyber-attacks**, to evaluate their impact on the platoon through model-based attack injection techniques exploring the digital twin e.g. [58]. In such situations, the platoon should be able to react quickly to prevent collisions.

## CRediT authorship contribution statement

**Maurizio Palmieri:** Conceptualization, Methodology, Software, Investigation, Simulation configuration, Simulation results analysis, Writing – original draft. **Christian Quadri:** Conceptualization, Methodology, Software, Investigation, Simulation configuration, Simulation results analysis, Writing – original draft. **Adriano Fagiolini:** Conceptualization, Methodology, Investigation, Simulation results analysis, Writing – original draft. **Cinzia Bernardeschi:** Conceptualization, Methodology, Investigation, Simulation results analysis, Writing – original draft.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

This work has been partially supported by the European Union by the Next Generation EU project ECS00000017 'Ecosistema dell' Innovazione' Tuscany Health Ecosystem (THE, PNRR, Spoke 3: Advanced technologies, methods and materials for human health and well-being) and partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence).

## References

[1] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, S. Tsugawa, Overview of platooning systems, in: Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012), 2012.

[2] C. Gomes, C. Thule, D. Broman, P.G. Larsen, H. Vangheluwe, Co-simulation: a survey, ACM Comput. Surv. 51 (3) (2018) 1–33, http://dx.doi.org/10.1145/3179993.

[3] J. Fitzgerald, P.G. Larsen, K. Pierce, Multi-Modelling and Co-Simulation in the Engineering of Cyber-Physical Systems: Towards the Digital Twin, Springer International Publishing, Cham, 2019, pp. 40–55.

[4] D.M. Botín-Sanabria, A.-S. Mihaita, R.E. Peimbert-García, M.A. Ramírez-Moreno, R.A. Ramírez-Mendoza, J.d.J. Lozoya-Santos, Digital twin technology challenges and applications: A comprehensive review, Remote Sens. 14 (6) (2022) http://dx.doi.org/10.3390/rs14061335, URL https://www.mdpi.com/2072-4292/14/6/1335.

[5] S. Mihai, M. Yaqoob, D.V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R.V. Prasad, H. Venkataraman, R. Trestian, H.X. Nguyen, Digital twins: A survey on enabling technologies, challenges, trends and future prospects, IEEE Commun. Surv. Tutor. 24 (4) (2022) 2255–2291, http://dx.doi.org/10.1109/COMST.2022.3208773.

[6] M. Palmieri, C. Quadri, A. Fagiolini, G.P. Rossi, C. Bernardeschi, Co-simulated digital twin on the network edge: the case of platooning, in: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2022, pp. 613–618, http://dx.doi.org/10.1109/WoWMoM54355.2022.00096.

[7] H. Van der Auweraer, J. Anthonis, S. De Bruyne, J. Leuridan, Virtual engineering at work: the challenges for designing mechatronic products, Eng. Comput. 29 (3) (2013) 389–408.

[8] K. Pierce, C. Gamble, D. Golightly, R. Palacín, Exploring human behaviour in cyber–physical systems with multi-modelling and co-simulation, in: E. Sekerinski, N. Moreira, J.N. Oliveira, D. Ratiu, R. Guidotti, M. Farrell, M. Luckcuck, D. Marmsoler, J. Campos, T. Astarte, L. Gonnord, A. Cerone, L. Couto, B. Dongol, M. Kutrib, P. Monteiro, D. Delmas (Eds.), Formal Methods. FM 2019 International Workshops, Springer International Publishing, Cham, 2020, pp. 237–253.

[9] L.I. Hatledal, R. Skulstad, G. Li, A. Styve, H. Zhang, Co-simulation as a fundamental technology for twin ships, Model. Identif. Control 41 (4) (2020) 297–311, http://dx.doi.org/10.4173/mic.2020.4.2.

[10] H. Feng, C. Gomes, C. Thule, K. Lausdahl, M. Sandberg, P.G. Larsen, The incubator case study for digital twin engineering, 2021, http://dx.doi.org/10.48550/ARXIV.2102.10390, URL https://arxiv.org/abs/2102.10390.

[11] C. Bernardeschi, P. Dini, A. Domenici, M. Palmieri, S. Saponara, Formal verification and co-simulation in the design of a synchronous motor control algorithm, Energies 13 (16) (2020) http://dx.doi.org/10.3390/en13164057, URL https://www.mdpi.com/1996-1073/13/16/4057.

[12] C. Bernardeschi, A. Domenici, A. Fagiolini, M. Palmieri, Co-simulation and formal verification of co-operative drone control with logic-based specifications, Comput. J. (2021) http://dx.doi.org/10.1093/comjnl/bxab161.

[13] Eureka network. URL https://www.eurekanetwork.org/.

[14] M. Williams, Prometheus-the European research programme for optimising the road transport system in Europe, in: IEE Colloquium on Driver Information, 1988.

[15] R. Rajamani, Han-Shue Tan, Boon Kait Law, Wei-Bin Zhang, Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons, IEEE Trans. Control Syst. Technol. 8 (4) (2000) 695–708, http://dx.doi.org/10.1109/87.852914.

[16] ENSEMBLE Project, The project, 2022, [Online; accessed 11-January-2022]. URL https://platooningensemble.eu/project.

[17] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev, C. Krupitzer, An overview on approaches for coordination of platoons, IEEE Trans. Intell. Transp. Syst. (2021) 1–17, http://dx.doi.org/10.1109/TITS.2021.3115908.

[18] M. Zabat, N. Stabile, S. Farascaroli, F. Browand, The Aerodynamic Performance of Platoons: A Final Report, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings qt8ph187fw, Institute of Transportation Studies, UC Berkeley, 1995, URL https://ideas.repec.org/p/cdl/itsrrp/qt8ph187fw.html.

[19] F.H. Robertson, F. Bourriez, M. He, D. Soper, C. Baker, H. Hemida, M. Sterling, An experimental investigation of the aerodynamic flows created by lorries travelling in a long platoon, J. Wind Eng. Ind. Aerodyn. 193 (2019) 103966, http://dx.doi.org/10.1016/j.jweia.2019.103966, URL https://www.sciencedirect.com/science/article/pii/S0167610519305641.

[20] B. McAuliffe, M. Lammert, X.-Y. Lu, S. Shladover, M.-D. Surcel, A. Kailas, Influences on Energy Savings of Heavy Trucks Using Cooperative Adaptive Cruise Control, SAE Technical Paper, SAE International, 2018, http://dx.doi.org/10.4271/2018-01-1181.

[21] R. Rajamani, Vehicle Dynamics and Control, in: Mechanical Engineering Series, Springer, Boston, MA, 2012, http://dx.doi.org/10.1007/978-1-4614-1433-9.

[22] D. Jia, K. Lu, J. Wang, X. Zhang, X. Shen, A survey on platoon-based vehicular cyber–physical systems, IEEE Commun. Surv. Tutor. 18 (1) (2016) 263–284, http://dx.doi.org/10.1109/COMST.2015.2410831.

[23] A. Bazzi, B.M. Masini, A. Zanella, I. Thibault, On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles, IEEE Trans. Veh. Technol. 66 (11) (2017) 10419–10432, http://dx.doi.org/10.1109/TVT.2017.2750803.

[24] C. Quadri, V. Mancuso, M.A. Marsan, G.P. Rossi, Edge-based platoon control, Comput. Commun. 181 (2022) 17–31, http://dx.doi.org/10.1016/j.comcom.2021.09.021.

[25] A. Virdis, G. Nardini, G. Stea, A framework for MEC-enabled platooning, in: IEEE Wireless Communications and Networking Conference Workshop (WCNCW), IEEE, 2019, pp. 1–6.

[26] X. Fan, T. Cui, C. Cao, Q. Chen, K.S. Kwak, Minimum-cost offloading for collaborative task execution of MEC-assisted platooning, Sensors 19 (4) (2019) 847.

[27] Y. Hu, T. Cui, X. Huang, Q. Chen, Task offloading based on Lyapunov optimization for MEC-assisted platooning, in: International Conference on Wireless Communications and Signal Processing (WCSP), 2019, pp. 1–5, http://dx.doi.org/10.1109/WCSP.2019.8928035.

[28] Alexios Lekidis, Faouzi Bouali, C-v2x network slicing framework for 5g-enabled vehicle platooning applications, in: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), 2021, pp. 1–7, http://dx.doi.org/10.1109/VTC2021-Spring51267.2021.9448769.

[29] M. Guiggiani, The Science of Vehicle Dynamics, Springer Netherlands, Pisa, Italy, 2014, p. 15.

[30] M. Burckhardt, Abs und Asr, Sicherheitsrelevantes, Radschlupf-Regel System, in: Lecture Scriptum, University of Braunschweig, Braunschweig, Germany, 1987.
[31] T.M.I. Hakim, O. Arifianto, Implementation of dryden continuous turbulence model into simulink for lsa-02 flight test simulation, in: Journal of Physics: Conference Series, Vol. 1005, IOP Publishing, 2018, 012017.
[32] M. Li, Y. Jia, J. Du, Lpv control with decoupling performance of 4ws vehicles under velocity-varying motion, IEEE Trans. Control Syst. Technol. 22 (5) (2014) 1708–1724.
[33] G. Tagne, R. Talj, A. Charara, Design and comparison of robust nonlinear controllers for the lateral dynamics of intelligent vehicles, IEEE Trans. Intell. Transp. Syst. 17 (3) (2015) 796–809.
[34] C.M. Kang, W. Kim, H. Baek, Cascade backstepping control with augmented observer for lateral control of vehicle, IEEE Access 9 (2021) 45367–45376.
[35] S. Pedone, A. Fagiolini, Racecar longitudinal control in unknown and highly-varying driving conditions, IEEE Trans. Veh. Technol. 69 (11) (2020) 12521–12535.
[36] S. Pedone, A. Fagiolini, Robust discrete-time lateral control of racecars by unknown input observers. http://dx.doi.org/10.1109/TCST.2022.3214054.
[37] Michele Segata, Renato Lo Cigno, Tobias Hardes, Julian Heinovski, Max Schettler, Bastian Bloessl, Christoph Sommer, Falko Dressler, Multi-technology cooperative driving: an analysis based on plexe, IEEE Transactions on Mobile Computing 22 (8) (2023) 4792–4806, http://dx.doi.org/10.1109/TMC.2022.3154643.
[38] P.A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using SUMO, in: IEEE International Conference on Intelligent Transportation Systems (ITSC), 2018.
[39] S. Santini, A. Salvi, A.S. Valente, A. Pescapé, M. Segata, R. Lo Cigno, A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios, IEEE Trans. Veh. Technol. 66 (3) (2017) 1985–1999, http://dx.doi.org/10.1109/TVT.2016.2585018.
[40] S. Hasan, J. Gorospe, S. Girs, A.A. Gómez, E. Uhlemann, Platoonsafe: An integrated simulation tool for evaluating platoon safety, IEEE Open J. Intell. Transp. Syst. 4 (2023) 325–347, http://dx.doi.org/10.1109/OJITS.2023.3271608.
[41] M.-A. Meyer, C. Granrath, G. Feyerl, J. Richenhagen, J. Kaths, J. Andert, Closed-loop platoon simulation with cooperative intelligent transportation systems based on vehicle-to-x communication, Simul. Model. Pract. Theory 106 (2021) 102173, http://dx.doi.org/10.1016/j.simpat.2020.102173, URL https://www.sciencedirect.com/science/article/pii/S1569190X2030112X.
[42] T. Blochwitz, M. Otter, J. Åkesson, M. Arnold, C. Clauß, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauß, D. Neumerkel, H. Olsson, A. Viel, Functional mockup interface 2.0: The standard for tool independent exchange of simulation models, in: Proc. of the 9th Intl. Modelica Conference, The Modelica Association, 2012, pp. 173–184, URL http://dx.doi.org/10.3384/ecp12076173.
[43] S.T. Hansen, C. Thule, C. Gomes, An fmi-based initialization plugin for into-cps maestro 2, in: C. Loek, M. Mieke (Eds.), Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops, Springer International Publishing, Cham, 2021, pp. 295–310.
[44] P.G. Larsen, J. Fitzgerald, J. Woodcock, P. Fritzson, J. Brauer, C. Kleijn, T. Lecomte, M. Pfeil, O. Green, S. Basagiannis, A. Sadovykh, Integrated tool chain for model-based design of cyber-physical systems: the INTO-CPS project, in: 2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data), 2016, pp. 1–6, http://dx.doi.org/10.1109/CPSData.2016.7496424.
[45] A. Junghanns, C. Gomes, C. Schulze, K. Schuch, R. Pierre, M. Blaesken, I. Zacharias, A. Pillekeit, K. Wernersson, T. Sommer, et al., The functional mock-up interface 3.0-new features enabling new applications, in: Modelica Conferences, 2021, pp. 17–26.
[46] A. Bazzi, A. Zanella, G. Cecchini, B.M. Masini, Analytical investigation of two benchmark resource allocation algorithms for lte-v2v, IEEE Trans. Veh. Technol. 68 (6) (2019) 5904–5916, http://dx.doi.org/10.1109/TVT.2019.2909438.
[47] ETSI GS MEC, Multi-Access Edge Computing (Mec); Framework and Reference Architecture, Tech. rep., 2019.
[48] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: A survey of the emerging 5 g network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681.
[49] J. Fitzgerald, C. Gamble, R. Payne, B. Lam, Exploring the cyber–physical design space, INCOSE Int. Symp. 27 (1) (2017) 371–385, http://dx.doi.org/10.1002/j.2334-5837.2017.00366.x, arXiv:https://incose.onlinelibrary.wiley.com/doi/pdf/10.1002/j.2334-5837.2017.00366.x. URL https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2017.00366.x.
[50] M. Palmieri, H.D. Macedo, Automatic generation of functional mock-up units from formal specifications, in: International Conference on Software Engineering and Formal Methods, Springer, 2019, pp. 27–33.
[51] Simulink®, www.mathworks.com/products/simulink.
[52] Y. Yao, L. Rao, X. Liu, X. Zhou, Delay analysis and study of ieee 802.11p based dsrc safety communication in a highway environment, in: 2013 Proceedings IEEE INFOCOM, 2013, pp. 1591–1599, http://dx.doi.org/10.1109/INFCOM.2013.6566955.
[53] M. Sepulcre, M. Gonzalez-Martín, J. Gozalvez, R. Molina-Masegosa, B. Coll-Perales, Analytical models of the performance of ieee 802.11p vehicle to vehicle communications, IEEE Trans. Veh. Technol. 71 (1) (2022) 713–724, http://dx.doi.org/10.1109/TVT.2021.3124708.
[54] C.M. Legaard, D. Tola, T. Schranz, H. Macedo, P.G. Larsen, A universal mechanism for implementing functional mock-up units, in: 11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications. SIMULTECH, 2021.
[55] V. Mancuso, P. Castagno, M. Sereno, M.A. Marsan, Modeling mtc and htc radio access in a sliced 5 g base station, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 2208–2225, http://dx.doi.org/10.1109/TNSM.2020.3034688.
[56] C. Thule, C. Gomes, K.G. Lausdahl, Formally verified fmi enabled external data broker: Rabbitmq fmu, in: Proceedings of the 2020 Summer Simulation Conference, SummerSim '20, Society for Computer Simulation International, San Diego, CA, USA, 2020.
[57] G. Nardini, D. Sabella, G. Stea, P. Thakkar, A. Virdis, Simu5g–an omnet++ library for end-to-end performance evaluation of 5 g networks, IEEE Access 8 (2020) 181176–181191, http://dx.doi.org/10.1109/ACCESS.2020.3028550.
[58] C. Bernardeschi, A. Domenici, M. Palmieri, Formalization and co-simulation of attacks on cyber–physical systems, J. Comput. Virol. Hack. Tech. 16 (1) (2020) 63–77.