2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM) | 978-1-6654-0876-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/WoWMoM54355.2022.00096

Co-simulated Digital Twin on the Network Edge: the case of platooning

Maurizio Palmieri[1], Christian Quadri[2], Adriano Fagiolini[3], Gian Paolo Rossi[2], Cinzia Bernardeschi[1]

[1]*Department of Information Engineering*, *University of Pisa*, Pisa, Italy
[2]*Computer Science Department*, *University of Milano*, Milan, Italy
[3]*MIRPALab, Department of Engineering*, *University of Palermo*, Palermo, Italy

***Abstract*—This paper presents an approach to create high fidelity Digital-Twin models for distributed multi-agent cyber-physical systems based on the combination of simulating components, generated from different modeling languages, each tailored for the specific domain of the subsystem. The approach specifically addresses the wireless communication domain, exploiting a Python module as a simulating component to evaluate the impact of network delay among the distributed elements of the system under analysis. A case study with a platoon of four vehicles following a leading car, all modeled in Simulink, is used to show the applicability of the approach, allowing the comparison between a Vehicle-to-Vehicle communication against a centralized multi-access edge computing paradigm.**

***Index Terms*—Digital Twin, co-simulation, Edge computing, vehicle platoon**

I. Introduction

One of the challenges emerged in recent years is to improve the road traffic and the long-distance transportation optimizing several factors such as safety, costs, risks and environmental sustainability by leveraging on autonomous or semi-autonomous driving systems, like the *platooning* systems.

Since 1987 platooning of vehicles has been one of the objectives of research in autonomous driving: in that year the Eureka [1] Project PROMETHEUS (PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety) [2] was started. Recently, progress has been made introducing *LIDAR* sensors and implementing algorithms like *object tracking* and *road and line detection* [3]. The combination of hardware and software is crucial in the Automated Driving System (ADS) area, increasing robustness to make the system safer and more efficient.

Platooning relies on closed-loop coordination among vehicles to achieve stability and short inter-vehicle distance. Data collected from on-board sensors are sent to other platoon members through communication network. Traditionally, platoon communications are based on V2V (Vehicle-to-Vehicle) scheme, which allows vehicles to directly exchange sensors data to other vehicles. Recently, controlling platoons using 5G network and edge computing has been proposed [4]–[6]. These approaches move the platoon controller to the edge relaying on low latency and reliable 5G communication.

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 movement of the vehicle and the kinematic model of the vehicle. When the model of a CPS is complex, simulation is a very important technique in the development process, as it allows the detection of any problems that may occur in the early stage of design. The different subsystems of a CPS may need to be modeled with different languages and tools, so it is useful to exploit a co-simulation approach that enables the heterogeneous simulation using multiple tools [7]. For example, co-simulation has been used by some of the authors in [8] for space coverage tasks of drones.

A digital twin is a virtual replica of physical processes created and maintained in order to gather insights about its physical counterpart [9]. A digital twin can also be used for monitoring, diagnostics and predictive maintenance. It provides risk-free environment where it is possible to investigate optimal operational conditions. In order to increase the fidelity of the digital replica, network communications among the different components of the system should be included in the digital twin.

The main contribution of this paper is the application of co-simulation technique to create a digital twin of a vehicle platoon, where a tool that specifically addresses the physical part of the system is coupled with a tool for the network connection that takes into account different traffic scenarios and different protocols.

The remaining of this paper is organized as follows: Section II introduces base information on the selected case study, namely the car platooning, Section III provides the different models used for the co-simulation and the architectures of the combined simulation units, Section IV shows and discuss the obtained results. Finally Section V concludes this paper, hinting some future extensions of this work.

II. Platooning Background

Platooning offers the opportunity to organize fleets of vehicles in groups traveling at short inter-vehicle distance reducing fuel consumption and enhancing road utilization. Maintaining platoon stability in a 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 [10]. In general, platoon coordination requires an intertwined system

978-1-6654-0876-9/22/$31.00 ©2022 IEEE
DOI 10.1109/WoWMoM54355.2022.00096

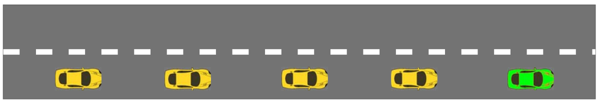


Fig. 1: Platoon example

between on-board sensors/actuators and communication protocols.

The main goal of a platoon control system is to maintain a specific inter-vehicle space and to guarantee *string stability* irrespective of the presence of external perturbation. The control is realized through specific control laws suitably designed to provide the vehicles with driving instructions to maintain the platoon stability. One of the most popular control law is Cooperative Adaptive Cruise Control (CACC) which is the result of the PATH project [11]. The goal of CACC is to maintain fixed inter-vehicular distance whatever is the platoon speed. The execution of CACC by a vehicle specifies the acceleration instruction the vehicle has to implement to maintain platoon stability. For each vehicle, CACC requires the data of the preceding and leader cars, in addition to the data of the vehicle itself. Data exchange among vehicles is performed through network communication protocols, as it is discussed in the following.

V2V communications is normally used to support inter-vehicle communication [12], through IEEE 802.11p and side link (PC5) 3GPP C-V2X [13]. Using these solutions, the platoon is managed in a distributed fashion and can operate autonomously without any supporting infrastructure, i.e. even in network out-of-coverage scenarios. However, the lack of a controlled and coordinated access to the wireless channel results in interference which leads to communication delays potentially compromising an effective platoon coordination. 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 in extra delays.

The emergence of 5G radio access networks (RANs) and dedicated network slices to the automotive domain in particular, have created opportunity to centralize the control of a platoon based on cellular communications and edge computing [14]. In this new setting, all radio communications are efficiently managed by the base stations, which can rely on effective MAC layer, by reducing inter-vehicle interference and shadowing effects. Besides, the system easily supports long platoons and even multi-platoon compositions. In edge assisted platoon, the information about the vehicle's state is sent to the platoon controller which is a network function running on an edge server. All computations are performed on the edge and results are sent back to the vehicle through RAN. More details about edge assisted platooning are available in [5].

III. Co-simulation modeling

Co-simulation can be used to study the behavior of complex CPS, such as vehicle platoons. The Functional Mockup Interface (FMI) [15] 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. 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, developed by the INTO-CPS Association, a result of the INTO-CPS project [16].

The co-simulation approach based on FMI standard is different from other modular approaches, e.g. PLEXE [17] 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.

The following subsections explain 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 case of Edge architecture and in case of V2V communications.

A. CACC FMU implementation

The instance of a typical CACC [11] with 5 cooperating vehicles is illustrated in Fig. 1. The mathematical formalization of the CACC is described by the following set of equations:

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

where the involved variables have the following meaning:

- $\boldsymbol{x_i}$ is the position of the *i*-th vehicle, $\boldsymbol{\dot{x}_i}$ is its speed and $\boldsymbol{\ddot{x}_i}$ is the acceleration; in particular $\boldsymbol{\dot{x}_0}$ is the speed of the platoon leader, and $\boldsymbol{\ddot{x}_0}$ is its acceleration;
- $\boldsymbol{x_{i-1}}$ is the position of the preceding vehicle, $\boldsymbol{\dot{x}_{i-1}}$ is its speed and $\boldsymbol{\ddot{x}_{i-1}}$ 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};
- $\boldsymbol{\dot{\varepsilon}_i}$ is the delta speed between the *i*-th vehicle and the preceding one;
- C_1 is the weighting factor between the acceleration of the leader and the acceleration of the preceding vehicle;
- $\boldsymbol{\xi}$ is the damping ratio;
- $\boldsymbol{\omega_n}$ is the controller bandwidth;
- d_{des} is the desired distance among the vehicles;
- $l_{i-1}$is the length of the vehicle;

These equations have been easily expressed in C code and then the FMU has been generated with PVSio-web [18]. The resulting FMU has 7 input variables ($\boldsymbol{x_i}$, $\boldsymbol{\dot{x}_i}$, $\boldsymbol{x_{i-1}}$, $\boldsymbol{\dot{x}_{i-1}}$, $\boldsymbol{\ddot{x}_{i-1}}$, $\boldsymbol{\dot{x}_0}$, $\boldsymbol{\ddot{x}_0}$), uses 5 parameters ($\boldsymbol{C_1}$, $\boldsymbol{\xi}$, $\boldsymbol{\omega_n}$, $\boldsymbol{d_{des}}$, $\boldsymbol{l_{i-1}}$) and produces 1 output ($\boldsymbol{\ddot{x}_{i\_des}}$).

B. Platoon FMU implementation

The MATLAB/Simulink suite [19] is one of the most renowned and qualified environment supporting block-based modeling. 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 low-pass filter macroscopically modeling the dynamic response of the car.

The kinematic model takes as input the acceleration *a* of the car and produces the updated speed *v* and position *x*. The equations (2) 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:

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

The Low Pass Filter takes as input the desired acceleration generated by the CACC algorithm and produces a First Order system response of the actual acceleration, that reaches the desired one through time.

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 single kinematic 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. Figure 2 shows the model for a 5-vehicle platoon, where 4 cars (violet background) follow a leading one (green background), whose acceleration is chosen as a sine wave signal. 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 vehicle) along with 4 more outputs that represent the distance between all couples of subsequent cars in the platoon. As a final remark on the model, Simulink provides a simple feature to export the model as an FMU that can be used to create an FMU with the same number of inputs, parameters and outputs of the model.

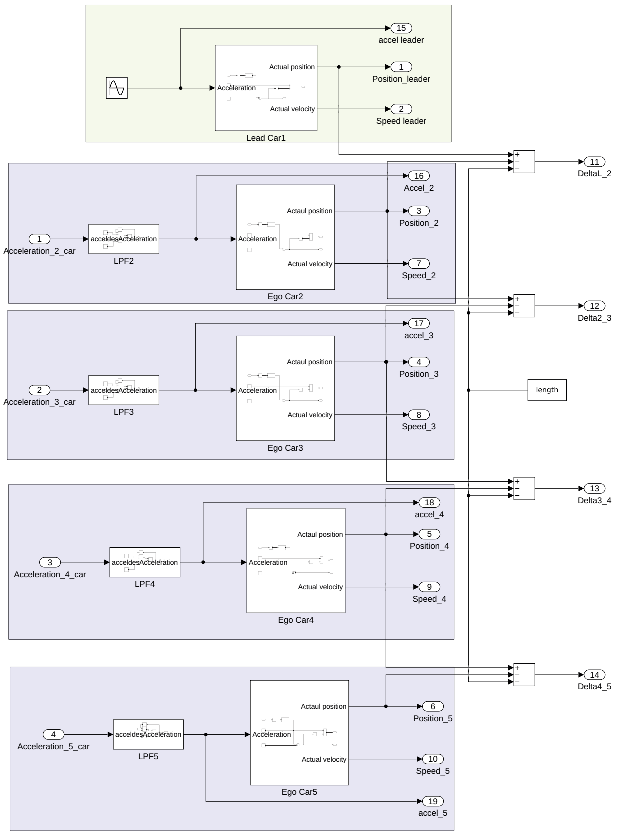


Fig. 2: Simulink Platoon model

C. General purpose delay FMU implementation

In order to introduce delays in the V2V scenario, a simple FMU sampling and holding its input value for a random delay has been implemented. The delay computation depends on a fixed number and a random number with exponential distribution. Such an FMU can be used for (i) a fixed delay, by setting a very low mean value for the average exponential distributed value, (ii) an exponentially distributed delay, by setting 0 to the fixed value and (iii) a combination of both the previous ones. Having only one input and one output, this FMU can be used as a general purpose delay because it can be added between any connection among FMUs in any co-simulation scenario.

D. Edge Network FMU implementation

The edge network is modeled by extending the standalone discrete simulator presented in [5] to support FMI standard using UniFMU [20]. This module is in charge of simulating the whole edge network including the user equipment (UE). The module consists of three sub-modules: (i) the network application running on-board of each vehicle, which periodically read the data from the on-board sensors and send them to the network; (ii) the radio access network (RAN) which simulates the base stations and manages the handover; and (iii) the

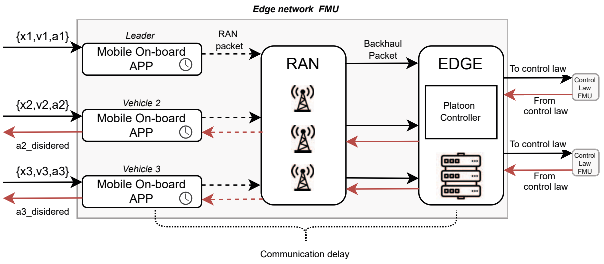


Fig. 3: Edge network FMU model

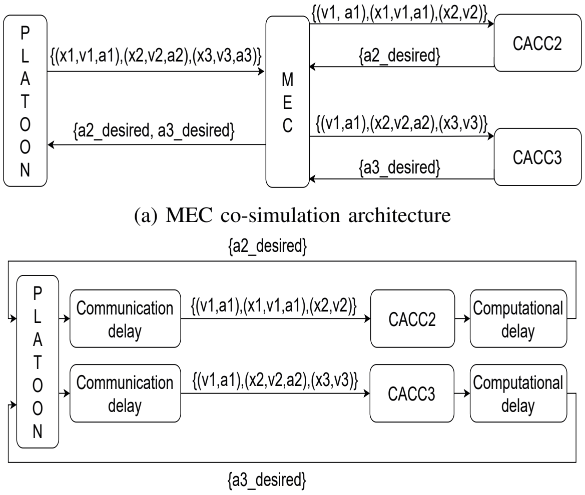
edge computing facility which is responsible for simulating the computational resources at the edge and the platoon controller edge application. In Fig. 3 we show the schema of edge network module. The black arrows represent the uplink data flow carrying all the information of on-board sensors, while the red arrows refer to the downlink flow for delivering the desired acceleration. The platoon controller, in 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 because more network nodes are involved in the communication between vehicles and edge controller. To model this aspect we use a pair of independent random variables, one for uplink and the other for downlink, configurable through FMU parameters. Each random variable can be defined as a simple random variable as well as a chain of independent random variables. This features gives the user the opportunity to model the edge network delays at different level of granularity.

E. Co-simulation architectures

Figure 4 shows the comparison between the two adopted architectures; in both cases, for the sake of simplicity, a simple platoon of three vehicles is shown. Moreover the figure uses a compact notation where *aX,xX and vX* stand for, respectively, acceleration, position and speed of the vehicle X in the platoon, considering vehicle number 1 as the leader. Figure 4a shows the architecture for the co-simulation scenario with the MEC infrastructure that involves, in the general case, $N+2$ FMUs: the one generated from Simulink, explained in Section III-B, the one generated from UniFMU, explained in Section III-D and N FMUs generated with PVSio-web, explained in Section III-A, where N is the number of following vehicles.

Figure 4b shows the logical architecture for the co-simulation scenario with the Vehicle-to-Vehicle communications that involves $3N+1$ FMUs: the one generated from Simulink that is exactly the same used for the other scenario, N FMUS, each representing the CACC algorithm executed on one car of the platoon, implemented as explained in Section III-A, N FMUs that introduce the delay for the communications and N FMUs that introduce the computational delay required to execute the CACC algorithm, where N is again the number of following vehicles.



(a) MEC co-simulation architecture

(b) V2V co-simulation architecture

Fig. 4: Comparison of logical co-simulation architectures

Parameter	Value
Platoon size	5 cars
Target distance	10m
C_1	0.5
ξ	1
ω_n	0.2
Vehicle length	4m
V2V Communication average delay	80ms
V2V Computation average delay	10ms
MEC fixed delay	10ms
MEC exponential average delay	5ms

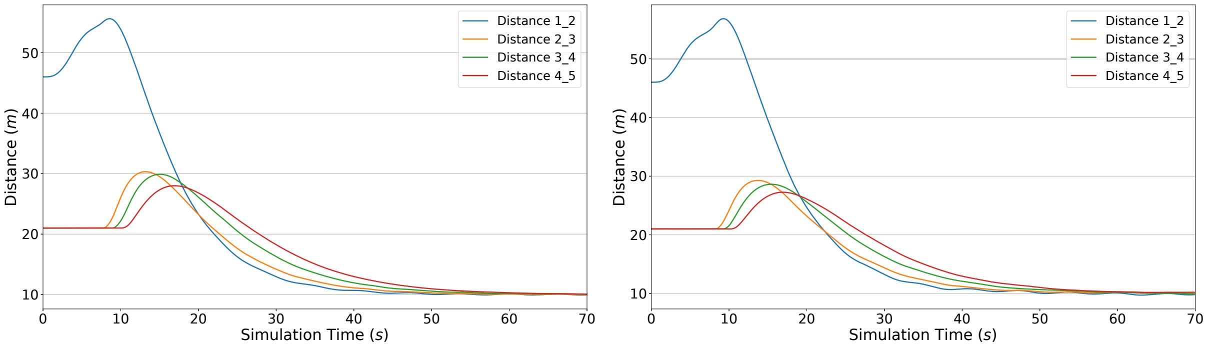
TABLE I: Parameters values.

IV. Co-simulation Results

Both co-simulation architectures 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 scenarios; it is possible to use lower step-size but it will lead to an increase of the required time to run the co-simulation. In Table I we report the main parameters of our simulation scenarios. For V2V scenario we consider a much higher communication delay than the typical bound of the IEEE 802.11p standard [21] to simulate a case of high interference channel leading to high number of retransmissions.

Figure 5 shows the comparison of the distances between two subsequent vehicles in the platoon. It is possible to notice that in both cases the distances between all the cars perfectly align to the target distance, i.e to 10 meters.

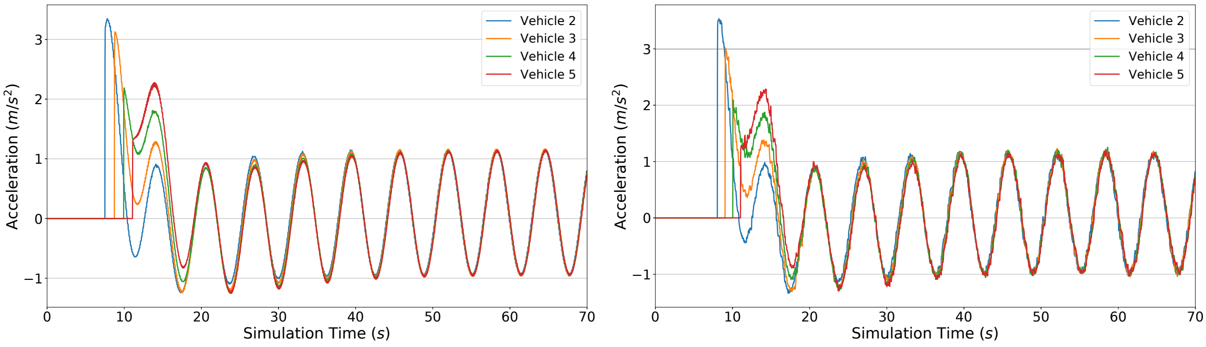
Figure 6 shows the comparison of the accelerations of the 4 following vehicles. It is possible to notice that the MEC infrastructure provides more stable acceleration values than the V2V approach.



(a) MEC co-simulation results.

(b) V2V co-simulation results.

Fig. 5: Comparison of distances.



(a) MEC co-simulation results.

(b) V2V co-simulation results.

Fig. 6: Comparison of acceleration.

Finally, Fig. 7a shows the comparison of the acceleration limited to the vehicle 2 and 5. It is possible to notice that the string stability is obtained and, as expected, the last car of the platoon exhibits lower oscillation during the initial maneuver. Figure 7b compares the distances between the leading car and the first follower (namely car number 2) and the distances between the last two cars (namely cars number 3 and 4) in V2V and MEC scenarios. It is possible to notice that in the V2V scenario the car number 2 reacts faster than the same car in the MEC scenario, while the other way round happens for the distance between the last two cars of the platoon.

V. Conclusions

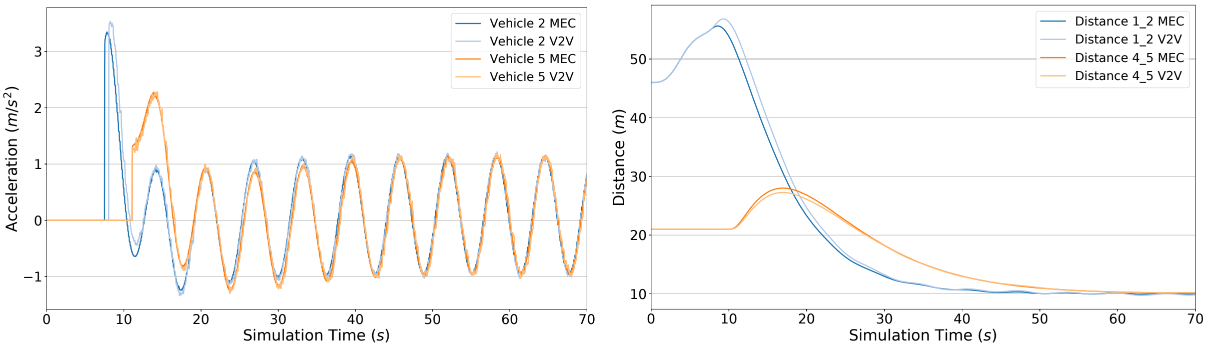
In this paper a digital twin of a platoon-based Vehicular CPS that includes an Edge based communication protocol for the exchange of data among the vehicles has been presented. In the digital twin the implementation of the CACC algorithm is coupled with a Simulink model for vehicle kinematics.

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 CACC algorithm implementation, *Simulink* for the physical plant and *UniFMU* for the network simulation.

The digital twin enabled the analysis of different scenarios in the context of a MEC or V2V based communication approach. In particular, the results show that the CACC algorithm performs in a correct manner, satisfying stability and communicating the desired acceleration to the physical system. Moreover, we have shown the benefits of the co-simulation approach, which has allowed us to easily prototype different scenarios by exploiting FMI standard and reusing already available models.

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

- **Enhancing the vehicle modeling using sensor data**, such as 3D LIDAR mapping, to give the vehicle the information of its precise position and to take action if a dangerous situation occurs.
- **Enhancing the network modeling, including channel behavior**, for example using OMNeT++ to model V2V or MEC communications.
- **Investigating on Design Space Exploration (DSE)**, to analyze different configuration parameters and find the combinations that perform better in terms of fuel/energy



(a) Acceleration comparison of vehicle 2 and vehicle 5.

(b) Distance of vehicle 2 to leader and vehicle 5 to 4.

Fig. 7: Cross-scenario comparison.

consumption and stability.

- **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. [22]. In such situations the platoon should be able to react quickly to prevent collisions.

References

[1] Eureka network. [Online]. Available: https://www.eurekanetwork.org/

[2] M. Williams, "Prometheus-the european research programme for optimising the road transport system in europe," in *IEE Colloquium on Driver Information*, 1988.

[3] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *CoRR*, vol. abs/1906.05113, 2019. [Online]. Available: http://arxiv.org/abs/1906.05113

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

[5] C. Quadri, V. Mancuso, M. A. Marsan, and G. P. Rossi, "Edge-based platoon control," *Computer Communications*, vol. 181, pp. 17–31, 2022.

[6] A. Lekidis and F. Bouali, "C-v2x network slicing framework for 5g-enabled vehicle platooning applications," in *VTC2021-Spring Workshops*. United States: IEEE, Feb. 2021, pp. (In–Press), 93rd Vehicular Technology Conference, VTC2021-Spring ; Conference date: 25-04-2021 Through 28-04-2021.

[7] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–33, 2018.

[8] C. Bernardeschi, A. Domenici, A. Fagiolini, and M. Palmieri, "Co-simulation and Formal Verification of Co-operative Drone Control With Logic-Based Specifications," *The Computer Journal*, 10 2021. [Online]. Available: https://doi.org/10.1093/comjnl/bxab161

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

[10] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev, and C. Krupitzer, "An overview on approaches for coordination of platoons," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2021.

[11] R. Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin 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, 2000.

[12] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, Firstquarter 2016.

[13] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, "On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 419–10 432, Nov 2017.

[14] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[15] T. Blochwitz, M. Otter, J. Åkesson, M. Arnold, C. Clauß, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauß, D. Neumerkel, H. Olsson, and 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. [Online]. Available: http://dx.doi.org/10.3384/ecp12076173

[16] S. T. Hansen, C. Thule, and C. Gomes, "An fmi-based initialization plugin for into-cps maestro 2," in *Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops*, C. Loek and M. Mieke, Eds. Cham: Springer International Publishing, 2021, pp. 295–310.

[17] M. Segata, R. Lo Cigno, T. Hardes, J. Heinovski, M. Schettler, B. Bloessl, C. Sommer, and F. Dressler, "Multi-Technology Cooperative Driving: An Analysis Based on PLEXE," *IEEE Transactions on Mobile Computing (TMC)*, 2 2022, to appear.

[18] M. Palmieri and 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.

[19] "Simulink$^{®}$," `www.mathworks.com/products/simulink`.

[20] C. M. Legaard, D. Tola, T. Schranz, H. Macedo, and 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.

[21] P. Fernandes and U. Nunes, "Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 91–106, 2012.

[22] C. Bernardeschi, A. Domenici, and M. Palmieri, "Formalization and co-simulation of attacks on cyber-physical systems," *Journal of Computer Virology and Hacking Techniques*, vol. 16, no. 1, pp. 63–77, 2020.