MISSIONE 4
ISTRUZIONE
RICERCA

FORESEEN

Deliverable D1.1-Report on the components and system definition of the CPS

Finanziato dall'Unione europea
NextGenerationEU

Ministero
dell'Università
e della Ricerca

Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA

Università
di Pisa

FORESEEN

FORmal m**E**thod**S** for attack d**E**t**E**ction in autonomous drivi**N**g systems

PRIN 2022 PNRR

Project number: P2022WYAEW
CUP: I53D23006130001

Deliverable D1.1: **Report on the components and system definition of the CPS**

Project Start Date: 30/11/2023 **Duration**: 24 months

Coordinator: *University of Pisa*

Deliverable No	D1.1
WP No:	WP1
WP Leader:	RU-PA
Tasks:	T1.1 - Leader: RU-PI T1.2 - Leader: RU-PA
Due date:	M1-6
Delivery date:	May 31, 2024
Authors:	RU-MI, RU-MOL, RU-PA, RU-PI

Dissemination Level:

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contents

List of Acronyms

ABS	Antilock Brake System
3GPP	3rd Generation Partnership Project
AV	Autonomous Vehicle
CACC	Cooperative Adaptive Cruise Control
CAN	Controller Area Network
CAV	Connected Autonomous Vehicle
COE	Cosimulation Orchestration Engine
CPS	Cyber Physical System
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
C-V2X	Cellular Vehicle to Everything
DSE	Design Space Exploration
DSRC	Dedicated Short Range Communication
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
LIN	Local Interconnect Network
MAC	Medium Access Control
MOST	Media Oriented System Transport
MPC	Model Predictive Control
PDR	Packet Delivery Rate
TSN	Time-Sensitive Networking
UAV	Unmanned Aerial Vehicle
V2E	Vehicle to Edge
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2V	Vehicle to Vehicle
VLC	Visible Light Communication

1. Introduction

The objective of this document is to define and describe a Cyber-Physical System (CPS) in terms of its main parts and interactions, to support the activities that must be performed in subsequent phases of the project. Moreover, it provides a brief explanation of the work done during the tasks T1.1 and T12. In particular, T1.1 was the knowledge acquisition task, where the objective was sharing expertise among the partners and creating a common background of knowledge in the multidisciplinary context of the project. Specifically, RU-PA's expertise on modeling vehicle dynamics and their interaction with the road and the environment, RU-PI's expertise on model-based design, validation and verification, RU-MI's expertise on network and edge computing technologies, and RU-MOL's expertise on applying formal method approaches to verify system properties. T1.2, instead, is the CPS modeling task, where the baseline model of the platoon taken into analysis should be provided.

In particular, the document provides information about the use case analysed in the project, the typical architecture of a CPS, the modelling approaches, the validation through simulation and the formal verification of properties.

2. Case Study

Vehicle platooning, a concept deeply rooted in the advancement of autonomous driving technology, refers to the method by which multiple vehicles travel closely together in a convoy, utilizing sophisticated communication and control systems to enhance safety, efficiency, and fuel economy. This approach is facilitated by Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, which allows the vehicles in the platoon to synchronize their movements, maintaining consistent speeds and minimal distances between them. The primary motivation behind vehicle platooning is to improve traffic flow and reduce fuel consumption. Furthermore, according to a study conducted by Alam et al., the implementation of platooning systems can lead to a substantial decrease in collision rates due to synchronized braking and acceleration [Alam 2015] and can contribute to congestion reduction. Indeed, by maintaining optimal speeds and reducing stop-and-go driving, platooning can help smooth traffic flows, which is particularly beneficial in urban areas with high traffic density. Research by Wang et al. demonstrated that traffic throughput could be increased by up to 50% in scenarios where platooning is widely adopted [Wang 2017].

From a control theory point of view, in general, controlling vehicle platoons presents greater complexity compared to traditional vehicle dynamics control. Indeed, the vehicles that constitute the platoon, can have different control strategies [Hu 2017] [Zheng 2016] and the entire platoon stability must be guaranteed (string stability) [Pauca 2023] [Mirabilio 2022] [Liu 2023].

3. Cyber-Physical Systems

System models are typically built with block-based languages, which describe a system as an assembly of functional blocks, each representing a possibly complex mathematical operation, interconnected by data flows. For example, Matlab is a commercial environment for the model-based development of CPS. It provides a graphical model editor and functions to generate demonstrative prototypes and to analyse relevant model properties.

Some works define ad-hoc solutions for the integration of different simulators, dedicated to different system components, e.g., the hardware and the software parts. For example, in [Bernardeschi 2018] Simulink and a simulator of logic specifications (the PVSio interpreter [Masci 2015] of the Prototype Verification System [Owre 1996]) have been integrated to simulate and formally verify properties of a pacemaker.

Farhat [Farhat 2018], instead, use Matlab and Simpack to compare the performance of mechatronically-driven railway vehicles' guidance and steering to that of a conventional vehicle.
A non-linear Simpack vehicle model with the specific mechatronic actuation and sensing, and a simplified linear control model in Matlab/Simulink are co-simulated.

Reinhart and Weissenberger [Reinhart 1999] address multibody simulation in the context of numerical control machines. The multibody model considers flexible machine structural components, guideways, feed drive dynamics and axis controllers together with the motion trajectory generation of the control.

In [Fagiolini 19], the architecture Robot Operating System (ROS)/Gazebo has been extended with the possibility of simulation of co-operative UAVs; in [Fagiolini 2020], Simulink/Gazebo tools are used for the validation of an approach to quadcopter control where wind disturbance is modeled by unknown exogenous inputs, exploiting the ROS middleware in the simulations.

3.1. Co-simulation

The design and development of CPS is increasingly complex due to inherent multidisciplinary issues. In CPS, simulation often takes the form of co-simulation. Co-simulation is the theory and technique that enable the global simulation of a coupled system via the composition of simulators.

Co-simulation has been applied extensively in different application fields. Among others, in [Bernardeschi 2020] a co-simulation approach is used for a brushless motor, with the electrical and mechanichal parts modeled in Simulink and the feedback linearization control modeled with a function written in C. In [Balasubramaniam 2022], a co-simulation open-source software for operation and planning studies of distributed energy resources has been presented. The software allows users to perform large-scale high-fidelity simulations for bulk power system (BPS) planning and operation. Papers [Hotzel 2021] and [Palmieri 2019] show how human performance models can be incorporated into models of CPSs. In particular, in [Hotzel 2021], a train driver model is coupled to models of the rolling stock and of the movement authority; in [Palmieri 2019] human-machine interfaces of an integrated clinical environment are considered.

In [Saponara 2022], co-simulation was applied to analyze Model Predictive Control systems for autonomous driving. This requires an accurate analysis of the interplay among three main components: the plant, the model predictive control algorithm, and the processor where the algorithm is executed. Co-simulation of the three components was used to determine if the controller running on the chosen hardware meets the time requirements determined by the response time of the plant. Satisfactory tradeoffs between algorithm complexity and processor performance could be studied.

In [Zizheng 2023], the co-simulation-based framework Vico for marine crane onboard operations is presented. Simulation and analysis of sub-systems is performed with different software tools and the framework enables the digitalization of marine operations.

In [Richart 2023], co-simulation is applied to co-operative mobile robots. Simulations for multi-robot systems are executed by independent tools, such as Gazebo for physics and mobility, ROS2 for software development, and ns-3 for communications and the networking infrastructure [Riley 2010]. The framework integrates such simulators and allows running experiments that combine all the involved robotic systems keeping the synchronization time between the simulators consistent.

In [Bernardeschi 2023], an approach to the formal verification of a variant of a well-known consensus protocol of UAVs is presented, using a co-simulation framework for system validation, and control parameter calibration is studied by design space exploration.

In [Cho 2022], Cho et al. present an advanced co-simulation platform that concurrently performs UAV simulation and wireless network simulation. This platform is taylored to the simulation of centrally controlled UAV swarms.

An extensive survey on co-simulation has been published by Gomes et al. [Gomes 2018], providing definitions of the fundamental concepts and a taxonomy of the literature based on the discrete events and continuous time computational models.

A paper by Fitzgerald et al. [Fitzgerald 2015] introduces foundational and process management issues for the design of CPSs and cites several methods, languages, and tools, using a two-wheel self-balanced personal transporter as an example.

The standard most widely used for co-simulation of dynamic systems is the Functional Mockup Interface (FMI) [Blochwitz 2011]. It has as key components, elements called Functional Mockup Units (FMUs), that represent a part of the system to be modeled. Each one of them is responsible for simulating the element that it is modeling and is independent from the other ones. During the co-simulation, FMUs that need to cooperate will be interfaced with each other in two possible ways:

- **Ad-Hoc:** Each FMU will be connected to another FMU throughout an ad-hoc Interface, as shown in Figure 1
- **Orchestrator**: The models are all connected, via an Interface, to a co-simulation orchestrator which is in charge of the communications between FMUs. A schema related to this example is shown in Figure 2.

Co-simulation is a flexible solution because every FMU is independent from the other, it can be developed in any language one desires. Another advantage is the fact that one can easily reuse components from other CPS in another case study. This is possible since FMUs are independent and for this reason it is also possible to easily modify a given component to extend it and add some new required features. During co-simulations it is possible to have, inside of our system, elements that refer to hardware components, software, or even some FMUs that model human interactions. In the first case we have Hardware-in-the-Loop co-simulation, while in the last case we have Human-in-the-Loop co-simulation. In this project, we use the Orchestrator solution for its generality.

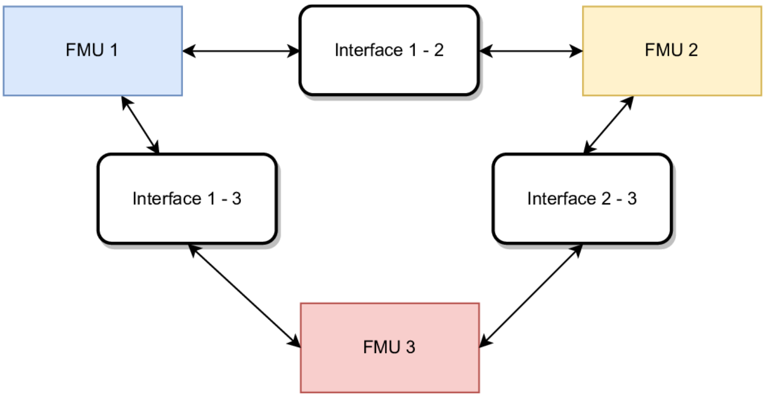

Figure 1: Schema Example of Ad-Hoc Co-simulation

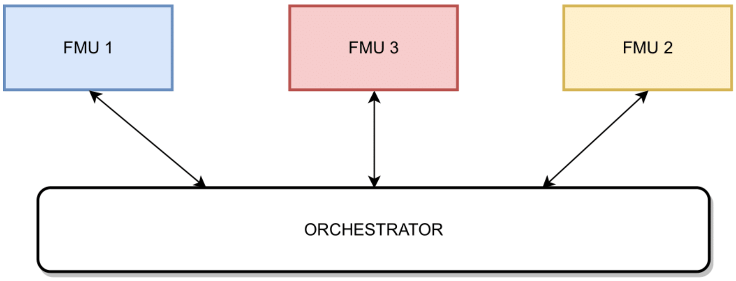

Figure 2: Schema Example of Co-simulation with Orchestrator

The Functional Mock-up Interface is a free standard that defines a container and an interface to exchange dynamic simulation models using a combination of XML files, binaries and C code, distributed as a ZIP file and hundreds of tools support this standard. Through modeling tools, one can generate the C Code that will interface itself with the methods that represent the behavior of the model, together with its set of inputs and outputs. The main component of the FMI standard is the FMU, which is in fact a zip file, with the extension **.fmu**, containing the information listed before.

It is important to note that each FMU has a solver who is in charge of running the simulation scripts.

Here we will see, in more detail, the elements inside of a Functional Mockup Unit:

- **modelDescription.xml**: it contains the description of the model that it is representing, in other words, it will specify the inputs, outputs and parameters of the FMU. For each variable there will be the name, the default value, if there is any, if the variable is fixed or can be modified during simulation, and finally, if the variable is an input, parameter or output.
- **C Code**: this refers to a set of functions that will be called during co-simulation in order to be able to interface with the solver of the FMU, which will be in charge of executing the scripts to simulate the behavior of the model.

The co-simulation paradigm is a Master-Slave, where the Master refers to the co-simulation engine, (the orchestrator), and the Slaves are the FMUs. Slaves can exchange information only throughout the Master and via a set of methods provided in the C Code inside of each FMU.

When a co-simulation starts, the Master will initialize the models by calling the functions **fmi2Instantiate** and **fmi2SetupExperiment**. Then it will call the **fmi2DoStep** which will increase the simulation time by a time step, that can be set at simulation design time, and then will pass data between FMUs via *set* and *get* methods called **fmi2Set**(*type*) and **fmi2Get**(*type*). At the end of the run, it will free the allocated resources by calling **fmi2Terminate** and **fmi2FreeInstance**.

In particular, the C methods are used to interface the Master with the solver of the specific FMU, meaning that the **fmi2DoStep** method will call another function that will simulate the behavior of the model, in the language used to develop such model. An example of simulation FMI-compliant is provided in Figure 3.

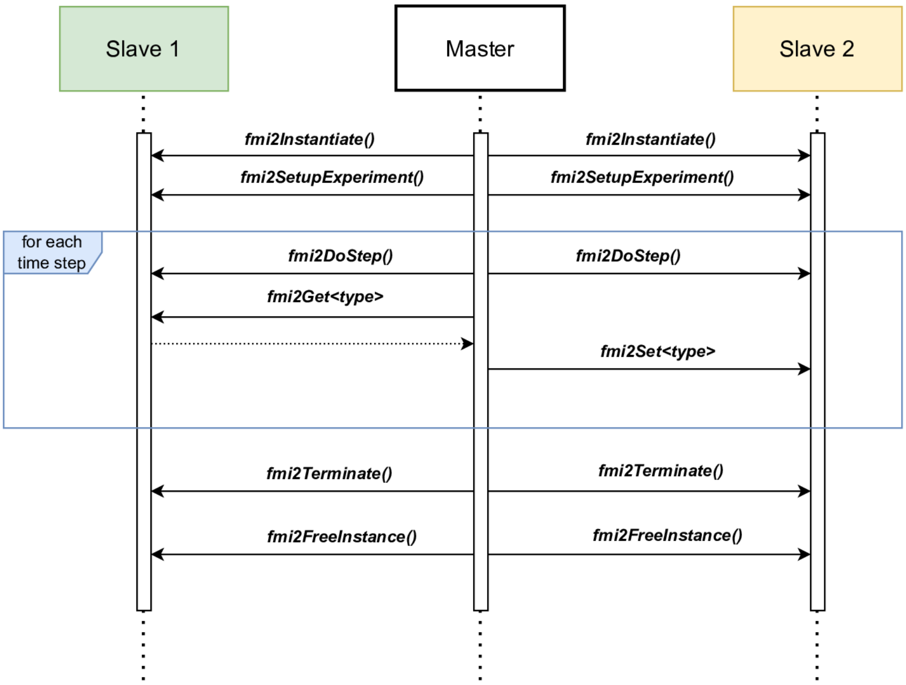

Figure 3: Example of Simulation FMI-compliant

In our work we will use the INTO-CPS framework [Larsen 2016] for the co-simulation of the cyber physical system.

INTO-CPS is a collection of tools developed to aid the development of Cyber-Physical Systems (CPS). It is one of the most widely used frameworks for co-simulation and it is FMI-compliant and the master is the Co-simulation Orchestration Engine (COE). INTO-CPS supports FMU obtained from models developed with Modelica, Simulink, Python and other various modeling tools.

Once one has the FMUs, it is possible to create Multi-Models by defining the components, their interconnections between each other and the values to be assigned to each parameter. The COE has a Graphical User Interface that allows to easily create and modify Multi-Models and run simulations, together with the possibility to visualize the results obtained from a run inside a graph and on **.csv** files that get generated after the co-simulation.

Another important feature that INTO-CPS provides is the Design Space Exploration tool (DSE), which is designed to solve optimization problems related to the system's parameters. This will be later used for the data gathering process, required to develop attack detection methods.

3.2. Network

Communication and networking play a fundamental role in CPSs. Different CPS applications impose varying requirements on their networks. For instance, healthcare CPS prioritize data integrity and privacy, while industrial CPS need deterministic latency and fault tolerance. Thus, network protocols must be tailored to meet these specific needs [Jawhar *2017*].

Focusing on connected autonomous vehicles (CAVs), they represent complex multi layered CPSs made of many components linked by different communication networks using specific network protocols. We can classify CAVs communication networks based on the scope of their communication.

Communication networks within autonomous vehicles (AVs) enable real-time data exchange among various subsystems such as sensors, actuators, and control units. Four key network technologies are commonly used in AVs: Controller Area Network (CAN), Local Interconnect Network (LIN), Media Oriented System Transport (MOST), and Ethernet. The *Controller Area Network* (CAN) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other, mainly focused on powertrain controllers. The *Local Interconnect Network* (LIN) is a lower-cost, lower-speed supplement to CAN, designed for simpler, non-critical applications within the vehicle. *Media Oriented System Transport* (MOST) is a high-speed multimedia network technology optimized for infotainment systems within vehicles. Finally, automotive *Ethernet* is increasingly being adopted in AVs due to its high-speed data transmission capabilities and versatility, and recent integration of time-sensitive networking (TSN) features for real-time communication [Ashjaei 2021]. In this project, we do not directly model the communication networks within the vehicles, but we take into account their vulnerabilities to model data and communication attacks.

Communication networks between vehicles enable the development of cooperative intelligent transport systems that reduce congestion and pollution, enhance travel and increase road safety. Traditionally, communications between vehicles have been realized through the dedicated short range communication (DSRC) approach, also named vehicle-to-vehicle (V2V), using radio frequencies (RF) IEEE 802.11p standard and visible light communication (VLC) IEEE 802.15.7 [Pranav 2019]. The DSRC approach does not require any support infrastructure and it operates in a distributed manner, however, V2V communication has several disadvantages. Its limited range restricts interactions to a few hundred meters, and network scalability issues arise with increasing vehicle density, leading to potential congestion. Moreover, V2V communication often requires a direct line of sight, making it less effective in obstructed environments. It also provides limited information, lacking data on broader traffic patterns or distant hazards. Security and privacy concerns are significant, as the system is vulnerable to hacking and spoofing.

In the last years, the 3rd Generation Partnership Project (3GPP) has proposed a new approach called Cellular Vehicle-to-Everything (C-V2X) which integrates vehicular communication in the mobile network standard. A significant milestone was reached in 2017 with the completion of LTE-V2X in Release 14, encompassing both direct and mobile network communications to deliver basic safety use cases. In 2020, Release 16 saw the completion of 5G-V2X, which similarly combined direct and mobile network communications to support advanced and automated driving use cases. The 3GPP consortium also integrates the ETSI MEC [ETSI 2024] (multi-access edge computing) standard architecture within the next generation of mobile network, enabling the opportunity to bring computational capacity to the edge of the networking, supporting latency-sensitive applications and services. The combination of C-V2X and MEC has opened a new era of innovation in the automotive vertical as the MEC4AUTO initiative of 5GAA, which proposes challenging use cases, such as intersection movement assist, safety of vulnerable road user through infrastructure sensors, and vehicle platooning.

Vehicle platooning is expected to significantly increase road utilization while reducing transport cost and driver fatigue. Platooning requires a tight integration between control algorithms and communication network, which has to offer stable and low-latency wireless link. Most proposals for managing a platoon of vehicle rely on DSRC for inter-vehicle communications [Jia 2016, Dressler 2019]. These studies highlight the necessity of integrating control and communication functions to maintain short inter-vehicle distances, particularly as the platoon size increases. Specifically, in [Wang 2023], a theoretical model of platoon message delays was presented, providing reference values for evaluating the performance of 802.11p technology. It was also observed that communication delays can cause increasing errors in inter-vehicle distance towards the end of the platoon. When this error remains stable, the platoon is considered string stable, making the primary challenge to control the distance between the leader and its first follower, with subsequent vehicles experiencing smoother variations and smaller control errors. These works also reveal that radio interference, shadowing, and multi-hop transmissions in V2V communications lead to string instability in long platoons. Conversely, broadband cellular networks like 5G offer better support and can even facilitate the virtualization of the platoon controller outside the platoon, such as at the edge of the cellular network [Quadri 2022, Virdis 2019] using the V2N (vehicular-to-network) communication approach.

In this project, we consider a CPS of a platoon of vehicles equipped with both V2V and V2N communication systems. For both systems we model delays and the message delivery probabilities as the main key quality of service parameters that affect the performance of platooning (see Section 3.2 of this document).

3.3. Formal Methods for CPSs

Autonomous systems are specific software for making decisions without human control, i.e., they are safety-critical systems. Autonomous systems are often embedded in a robotic system becoming a part of everyday life, interacting with humans, whether through physical interactions or other modes. For the above reasons, autonomous systems require robust development and verification approaches since a failure can cause harm to humans or even their death.

Autonomous vehicles, also known as self-driving or driverless vehicles, are specific autonomous systems equipped with advanced technologies enabling them to operate and navigate without human intervention. These vehicles leverage a combination of sensors, cameras, radar, lidar, GPS, and sophisticated software algorithms to perceive their surroundings, analyze the collected data, and make immediate decisions to manage the vehicle’s movements in real-time.

Autonomous vehicles and platooning are two interrelated advancements in vehicular technology that are transforming the landscape of modern transportation. While both concepts can exist independently, their combination offers significant enhancements in terms of safety, efficiency, and driving experience. Platooning refers to the coordinated control of a group of vehicles, typically using automated driving technologies to enable them to travel closely together at high speeds with minimal human intervention.

Given the safety-critical nature of vehicle platooning, formal methods have been extensively employed to ensure that these systems operate correctly under all possible conditions.

Formal Methods encompass rigorous approaches for the specification, development, and verification of software and hardware systems. They provide a rigorous foundation for modeling complex systems and proving properties about them, such as correctness, safety, and reliability. Common formal method techniques include model checking, theorem proving, and abstract interpretation. These techniques are invaluable in areas where system failure can have catastrophic consequences, such as aerospace, medical devices, and automotive systems.

Thus, researchers successfully used Formal Methods to accurately verify vehicle behavior in various situations, including, software correctness and reliability [Pola and Di Benedetto, 2019], driver profiling [Martinelli 2021], object recognition [Dokhanchi 2018], route planning [Mehdipour 2023] and response to unexpected scenarios [Krichen 2023], e.g., security breaches.

Mehdipour et al. [Mehdipour 2023] conducted a review of recent studies employing Formal Methods in the context of autonomous driving with particular attention to autonomous route planning. Their focus encompassed formal specifications for road rules, with a particular emphasis on temporal logic. Their review encompassed verification, monitoring, and control synthesis techniques derived from such specifications. Mehdipour et al. scope was limited to ego-centric approaches and system-level methodologies that examine the overall behavior of an autonomous vehicle, rather than specific software code within the vehicle. Additionally, they provided a critical discussion of the field, addressing ongoing challenges and proposing directions for future research. Finally, they observed how the expressivity of temporal logic formulas can be harnessed to formalize traffic laws.

Moreover, Krichen [Krichen, 2023] provides a comprehensive overview of the current state-of-the-art Formal Methods and validation techniques employed in the automotive industry for system security. Krichen delves into diverse Formal Method techniques, including Model Checking, theorem proving, and abstract interpretation, extensively used for analyzing and verifying the security properties of automotive systems. Moreover, this overview highlights the validation techniques applied to ensure the efficacy of security measures, encompassing penetration testing, fault injection, and fuzz testing. Additionally, the paper investigates the integration of Formal Methods throughout the automotive development lifecycle, spanning requirements engineering, design, implementation, and testing phases. It delves into the advantages and limitations of these approaches, considering factors such as scalability, efficiency, and applicability to real-world automotive systems.

In platooning landscape, it is crucial to ensure that vehicles maintain safe distances and respond promptly in emergencies. Therefore, platooning systems must be validated to guarantee reliable behavior. To achieve this, it is essential to establish clear validation strategies, such as formal verification and simulation.

Despite the critical importance of safety, much of the current research on longitudinal control units focuses on optimization and control methods. For instance, Mosbach et al. [Mosbach 2017] introduces a cooperative shared control driver assistance system that aids drivers in longitudinal vehicle control. Zhao et al. [Zhao 2018] propose robust longitudinal control units for each vehicle, except the leading one, based on an ideal swarm model, emphasizing collision avoidance without formal verification of the model. Li et al. [Li 2017] present a distributed H-Infinity control method for multi-vehicle systems. They use identical dynamic controllers and rigid formation geometry, focusing primarily on robustness and string stability rather than safety properties.

In addition to optimization and control methods, Rahman et al. [Rahman 2018] examine the longitudinal safety of platoon systems on dedicated highway lanes. This simulation-based analysis demonstrates that platoon systems significantly enhance longitudinal safety compared to the baseline condition.

Model checking, one of the widely used formal methods, involves creating a finite model of the system and exhaustively checking whether certain properties hold. This is typically done using temporal logic specifications. Theorem proving, on the other hand, involves constructing mathematical proofs to verify the correctness of systems. Abstract interpretation is another technique where an abstract model of the system is analyzed to conclude the actual system's behavior.

Peng et al. [Peng 2019] present a timed automata model of a vehicle platoon system to determine a minimal yet guaranteed safe distance between two vehicles under variable speed conditions. Unlike other models based on cooperative adaptive cruise control, this approach assumes no (Internet) communication among different vehicles or with the road system. Instead, it adopts a local perspective: each vehicle relies on its sensors to dynamically calculate and maintain a safe distance from the preceding vehicle in the platoon. The model checker UPPAAL is used to verify that the system does not deadlock and, most importantly, ensures safety by always avoiding crashes.

Similarly to the above approach, Rodonyi [Rödönyi, 2017] suggests a virtual spacing policy to replace unknown spacing policies in platoon systems, yet lacks formal verification of safety properties.

In addition to the above challenges, platooning vehicles are subject to interference from other agents, often malicious, affecting their operations to destabilize the platooning [Dadras 2018, Singh 2018]. Platooning attacks can be divided into several categories: information availability, integrity, authenticity, or confidentiality. For example, Vasconcelos et al. [Vasconcelos 2024] provide an overview of cybersecurity research works related to platooning vehicles, particularly highlighting the impact of security threats on the platooning applications and proposing strategies to mitigate them.

Mousavinejad et al. [Mousavinejad 2019] present a distributed attack detection mechanism. In their approach, each vehicle estimates the local leader's position and evaluates received information, proposing two recovery methods based on system state estimation. Other works, like Cao and Yin [Cao 2021] and Singh et al. [Singh 2020], suggest employing blockchain to establish secure platoons among vehicles for trust management and collaborative decision-making.

Basiri et al. [Basiri 2022] propose an attacker-detector game using a centralized detector to identify the best vehicles for sensor addition to detect the attack. Furthermore, several countermeasures against position falsification attacks using a proof location scheme are proposed by Boeira et al. [Boeira 2018] to avoid collisions by detecting false messages.

Conversely from the above works, our focus is on the security aspects of platooning. More in detail, we are interested in identifying vehicle platooning attacks using rigorous techniques, i.e., formal methods, with particular attention to model checking technique.

3. Platoon - overview

In the following a baseline model of the platoon is presented, where an ideal road and network is implemented. In this system there is a leading car that moves at a variable speed, depending on an acceleration that is a sinusoid function, which is followed by a platoon of 3 cars. Each car has a controller that is wirelessly connected with the other cars, leader included. The communication between vehicles is simple, a car that is behind another car receives from the preceding one its speed, position and acceleration, while every car receives from the leader its speed, position and acceleration, called in this case V_leader, x_leader and a_leader. Furthermore, each car will have to notify to its controller, locally, at which speed it is moving, and its actual position, obtaining in return the acceleration to have in order to obtain a steady state in which every car has the same speed of the leader, while being at a specific distance one to the other, in our case, 15 meters. The algorithm used to compute the desired acceleration of each simple car is the Cooperative Adaptive Cruise Control (CACC) algorithm, in which the desired acceleration is computed by each controller as follows:

$$a\_ego = C1 * a\_leader + (1 - C1) * a\_front - K1 * (V\_ego - V\_leader) - K2 * (x\_ego - x\_front + L)$$

In the formula, ego denotes any car except the leader, *front* is the immediately preceeding car, L is the safe distance. $C1$, K1 and $K2$ are control parameters. Figure 4 shows a platoon with the leader and three follower cars; the safe distance between the cars is15 meters.

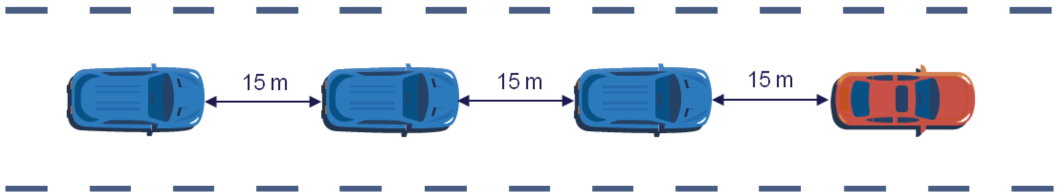

Figure 4: Visual representation of the platoon

3.1 Physical components

Following the reasoning made before, the considered vehicle dynamics model is presented. More specifically, according to the project's purpose, we will refer to the longitudinal dynamics. In this setting, we consider a car with rear traction traveling along a flat and straight road. Under the common assumptions of having tires with identical geometric and inertial characteristics and traction that can be generated instantaneously, the longitudinal dynamic behavior of the car is described by the so-called bicycle model [Guiggiani 2023]. Referring to such a model, by denoting with ω the speed of the virtual rear wheel and with u the longitudinal velocity of the vehicle, the following dynamic equations can be used

$$I_{\omega}\dot{\omega} = T - R_{\omega}(\dot{u})F_t(\omega, u, \dot{u})$$

$$m\,\dot{u} = F_t(\omega,\, u,\, \dot{u}) - F_a(u) - F_r(u)$$

where I_ω is the wheel's inertia along the rotation axis, m is the vehicle mass, T is the input driving torque, R_ω is the so-called effective tire radius, F_t is the rear traction force, F_a is the aerodynamic drag force, and F_r is the total rolling resistance force summing up the effects at all tires. Though the model appears to be very compact, its complexity is actually encoded in every single term. More precisely, the effective tire radius R_ω relates the rotational wheel velocity ω to the linear longitudinal wheel velocity as it moves through the contact patch of the tire with the ground [Rajamani 2012]. It can be shown that R_ω can be expressed according to the formula:

$$R_\omega = \frac{\sin\left(a\cos\left(\frac{R_d}{R}\right)\right)R}{a\cos\left(\frac{R_d}{R}\right)}$$

where R and R_d are the tire's undeformed and dynamic radii, respectively, the latter of which is obtained as follows:

$$R_d = R - \frac{F_z(\dot{u})}{K_t}$$

with F_z the vertical force acting on the rear tire and K_t its stiffness along the vertical direction. In turn, the vertical force F_z is obtained as a static term, depending on the geometry of the car and being present in conditions of constant longitudinal speed, and a dynamic one directly proportional to the longitudinal acceleration $\dot{u}$ and expressing the normal load transfer between front and rear axles during acceleration and braking phases. In particular, it holds

$$F_z = \frac{m(g\,a_1 + h\,\dot{u})}{l}$$

with l is the wheelbase, i.e. the distance between the rear and front axles, a_1 is the distance between the front axle and the vehicle's center of mass, and h is the height of the center of gravity from to the road. Based on this reasoning the effective tire radius is in the general case a function of the instantaneous longitudinal acceleration, i.e. $R_\omega = R_\omega\,\dot{u}$

Moreover, the rear traction force F_t is highly dependent on many factors, among which are the characteristics of the road and the type of interaction between the tire and the road asphalt. At the state-of-the-art only heuristic expressions are available, which consist of static models providing the instantaneous traction force F_t as a function of the wheel speed ω, longitudinal velocity u and acceleration $\dot{u}$, and other parameters to be ad-hoc identified. Similar reasoning holds for the expressions of the aerodynamic drag force F_a and rolling resistance force F_r. . Here, it is also assumed that the vehicle's pitching motion is limited and can occur only at the beginning of the acceleration or braking phase. This allows neglecting the effect of the suspensions on the vehicle's dynamic behavior, which is present only during short initial transients.

Furthermore, it can be assumed that the wheel's and longitudinal vehicle's speeds are available. Indeed, in modern cars, measurement of the wheel speed ω can be obtained from the Antilock Brake System (ABS), while that of the vehicle's longitudinal speed u can be provided by speedometer sensors, whose accuracy can be improved by fusion with GPS speed data [Walter 2013] and even LiDar measurement [Massa 2020].

The considered longitudinal vehicle dynamics model is simulated via the Vehicle Dynamics Blockset (Vehicle Dynamics Blockset, URL: https://it.mathworks.com/products/vehicle-dynamics.html#full-vehicle), a particular Simulink toolbox that allows to simulate the vehicle dynamics with a high-fidelity level. More specifically, the vehicle dynamics is simulated via the Vehicle 1DoF Double Track block (Vehicle Body 1DoF, https://it.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html) that implements a rigid two-axle vehicle body model to calculate longitudinal motion. It takes the longitudinal traction forces, generated at the tire-asphalt interfaces derived from the driving wheels, as inputs. The latter are simulated using the Longitudinal Wheel block (Longitudinal Wheel Block, URL: https://it.mathworks.com/help/autoblks/ref/longitudinalwheel.html) with traction forces simulated via the Pacejka's magic formula and rolling resistance following ISO 28580 standards. By varying the values of Pacejka's magic formula it was possible to simulate different road conditions.

3.2 Network

The communication network for the platoon is realized considering the V2V and V2N approaches and leveraging the co-simulation technique. As concern the wireless communication standards, we use IEEE 802.11p[1] as standard for V2V communication, while we consider 5G-NR[2] and ETSI MEC [ETSI 2024] standards for V2N communication type.

We adopt a similar design of the network components as in [Palmieri 2023] focusing on the main parameters that affect platoon communications, i.e. communication delay and packet delivery ratio. We exclude direct modeling the link bandwidth as both standards, 802.11p and 5G-NR, offer abundant link data rate compared to the one required by the platoon application which is roughly 80 kpbs. In particular, 802.11p standard offers a nominal bit rate of 6 Mbps using QPSK ½ modulation and coding scheme, while 5G-NR offers more that 10 Mbps in uplink and downlink even in highly degraded channel quality.

The V2V communication approach does not require any supporting infrastructure, and the platoon is managed in a fully distributed manner. Despite the deployment advantages, V2V solutions face several issues that hinder their scalability and effectiveness. Uncoordinated access to the radio channel is challenging because it increases communication delays due to channel access contention. Another aspect to consider in V2V communication is its high sensitivity to radio shadowing, particularly when large vehicles are present, resulting in significant packet loss. We model the communication delays and packet delivery ratio of V2V communication assuming the following conditions:

1. The Medium Access Control (MAC) layer operates using Enhanced Distributed Channel Access (EDCA, 802.11e) which implements QoS mechanism based on the application type. Platoon messages are classified as high-priority messages; thus, they do not compete with other less-critical application messages, such as infotainment.
2. As mentioned above, the access to the radio channels is uncoordinated and each vehicle within the platoon, as well as other vehicles that are not part of the platoon, compete for accessing the channel. The adoption of the CSMA/CA (carrier sense multiple access/collision avoidance) algorithm does not offer guarantees about queueing delay, i.e., the time a packet must wait until the radio channel is free. To model this aspect, we use the

---

[1] IEEE 802.11p: IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, available at https://standards.ieee.org/ieee/802.11p/3953/

[2] 3GPP TR 38.811 Study on scenarios and requirements for next generation access technologies

approach presented in [Yao 2013] which shows that the *exponential distribution is a good approximation to the MAC access delay*. By varying the λ parameter of the distribution we are able to model different scenarios having different vehicle densities.

3. We assume that the noise level is almost identical at each receiver antenna of the vehicles that are member of the platoon. Moreover, we assume that transmission power and the type of antennas are identical for each vehicle within the platoon. For these reasons, the packet delivery ratio (PDR) is mainly affected by the distance between the transmitter and receiver. To model PDR we use the analytical model presented in [Sepulcre 2022], which considers the PDR as a function of the transmission power and the distance between transmitter and receiver. This model gives us a high degree of flexibility to model a wide set of scenarios.

Regarding the V2N communication approach, we consider a two-hop communication between vehicles and the edge server where the platoon controller is deployed (see the following section). Moreover, being the wireless link between the vehicle and the base station asymmetric in terms of delay, we model the uplink and downlink communication paths separately as proposed in [Quadri 2022].

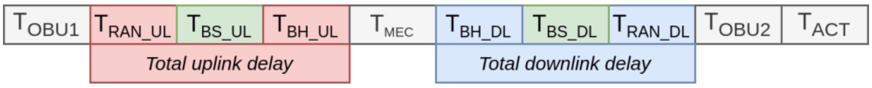

Figure 5: Model of the delay components of V2N communication approach.

In Figure 5, we report the model of the delay components of the communication between vehicles and edge server in both directions. The components in grey are delays caused by onboard computation (T_{OBU1} and T_{OBU2}), edge server computation of the control law (T_{MEC}, see next section for details), while T_{ACT} refers to the actuation lag due to physical inertia of the vehicle, as described in Section 3.1. The other delay components are directly induced by the mobile network. More specifically, the uplink is made of three parts:

1. T_{RAN_UL} represents the transmission time of a message between the vehicle and the base station that is serving it. This component models the time required to ask the base station for uplink transmission permission, the sending of the transmission grant (in downlink), the actual message transmission, and the acknowledge from the base station.
2. T_{BS_UL} is the time spent by the base station to receive, decode and process the message and send it to the edge server through the backhaul network. This component can also be used to model network congestion at base station level in uplink direction.
3. T_{BH_UL} is the transmission time of the message between the base station and the edge server. Being the backhaul network equipped with optical fiber offering high data-rate, the delay is mainly caused by the geographical distance between the base station and the edge server where the platoon controller is deployed. The modeling of this specific component allows us to model different edge network topologies and application deployments. A short delay models a deployment scenario where the platoon controller is deployed on edge server geographically close to the base station. On the contrary, a long delay models the scenario of a deployment on edge server at regional or even national level.

The downlink delay is modelled similarly:

1. T_{BH_DL} is the transmission time of the message between the edge server and the base station.
2. T_{BS_DL} is the time spent by the base station to receive, decode and process the message and send it to vehicle. Similarly to T_{BS_UL} component, this component can also be used to model network congestion at base station level in downlink direction.

3. T_{RAN_DL} represents the downlink transmission time, which involves the notification to the vehicle that a message is going to be transmitted, the actual transmission and the final acknowledgment.

All the network delay components are modelled as random variables offering the opportunity to model a board set of scenarios, by varying congestion levels, edge server geographic deployment and transmission delays.

3.3 Application

The application that controls the platoon is composed by three modules as depicted in Figure 6

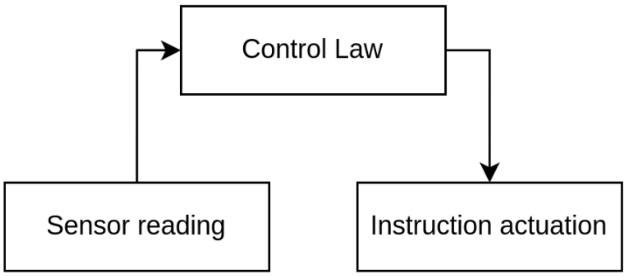

Figure 6: Application components.

Based on the selected communication approach we have different deployments of the application controlling the platoon.

In the case the V2V communication paradigm is used, all the application components are deployed on-board of each vehicle of the platoon and the control law (e.g., CACC) is computed by each vehicle independently using the data read from on-board sensors and the data received from the other vehicles through V2V communication network.

Conversely, in the V2N approach, the control law component is deployed on an edge server that collects data sent by all vehicles in the platoon, computes the control law and sends back the instruction to all vehicles.

In both communication approaches, the sensor reading is performed periodically.

3.4 Components interaction

The physical components and the network should be interconnected between each other in order to create the multi model to co-simulate. A general view to understand how this should be done can be seen in the two following figures.

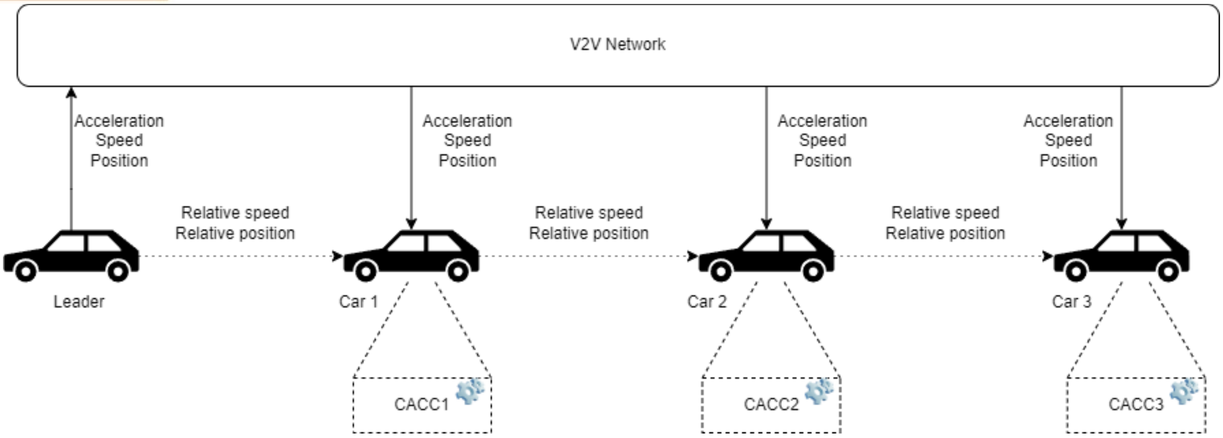

Figure 7: Vehicle to Vehicle general schema

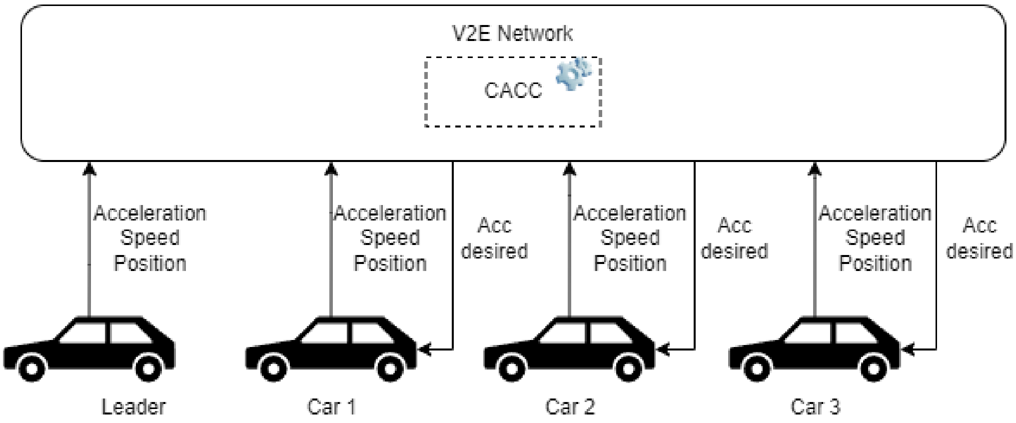

Figure 8: Vehicle to Edge general schema

The first image shows the platoon in a V2V scenario. It can be seen that the leader provide every other following car its acceleration, speed and position, while each vehicle, through the use of sensors, can obtained the values for the relative speed and distance from the preceding vehicle. Those values will be used by the Controller, which is running the CACC algorithm on board.

The setup for the V2E approach is similar, but vehicles does not need to compute values using sensors, since the decision making process is shifted inside of the network. In fact, the CACC now is running on cloud and the network provides the desired acceleration to the vehicles.

This general approaches have been brought to INTO CPS creating the multi-model, as shown in Figure 9 and Figure 10.

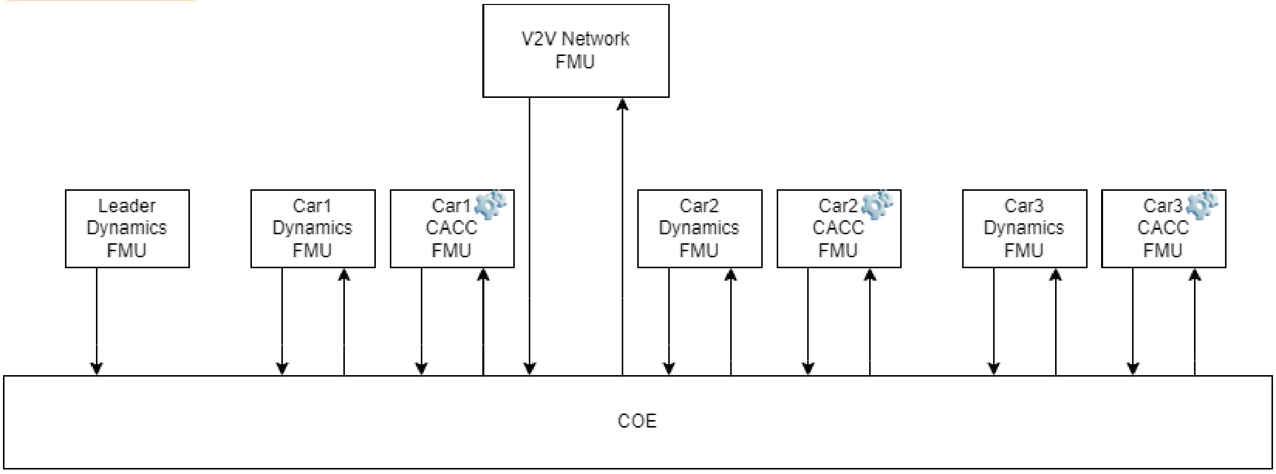

Figure 9: Co-Simulation schema for V2V

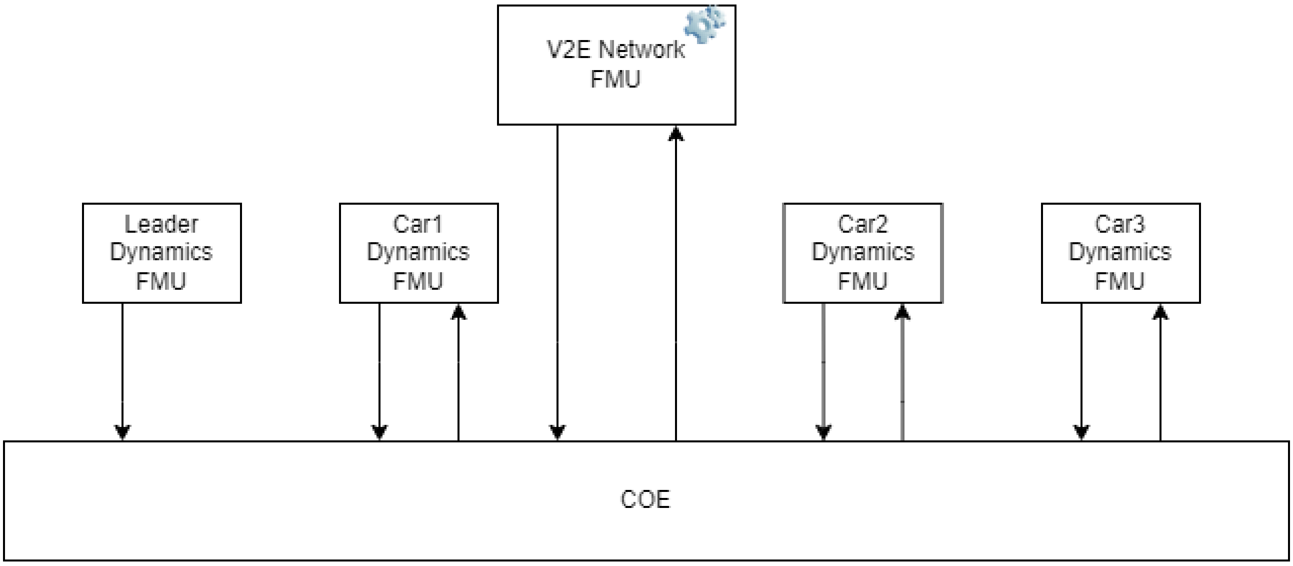

Figure 10: Co-Simulation schema for V2E

The first figre shows the multi model for the V2V configuration, while the latter figure refers to the V2E approach. It is important to notice that when running a co-simulation, the COE will be in charge of making each FMU evolve and, most importantly, will take care of the exchanges of data from component to component.

4. Roadmap for future work

This deliverable gives an overview of background on modelling a CPS and the application of formal methods for safety and security. The vehicle platooning application is used as running example. Connected and autonomous vehicles (CAVs) rely on an array of sensors to navigate and perceive their surroundings. Moreover, the software within the vehicle interprets the sensor data and issues commands to the actuators, which control the vehicle's movement. Consequently, any threats to sensors or actuators pose risks to the entire system’s security. The next steps will be the threats analysis and identification of critical physical devices, as well as the definition of attack scenarios.

5. Bibliography

[Alam 2015] Alam, Assed, et al. "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency." IEEE Control Systems Magazine, 35.6 (2015): 34-56

[Ashjaei 2021] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, Saad Mubeen, “Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities”, Journal of Systems Architecture, Volume 117, 2021, 102137, ISSN 1383-7621.

[Balasubramaniam 2022] Balasubramaniam, Karthikeyan and Plathottam, Siby and Yim, Sang-Il and Kang, Ning and Jhala, Kumarsinh and Bhattarai, Rojan and Zhao, Shijia, “Co-Simulation of Transmission and Distribution Systems—From Modeling to Software Development”, in IEEE Access, vol. 10, pp. 127061-127072, 2022, doi: 10.1109/ACCESS.2022.3224939

[Basiri 2022] Basiri, M. H., Pirani, M., L. Azad, N., and Fischmeister, S. Security-aware optimal actuator placement in vehicle platooning. Asian Journal of Control, 24(4):1557–1574, 2022.

[Bernardeschi 2018] C. Bernardeschi, A. Domenici, P. Masci. A PVS-Simulink Integrated Environment for Model-Based Analysis of Cyber-Physical Systems. IEEE Trans. Software Eng. 44(6): 512-533, 2018

[Bernardeschi 2020] Cinzia Bernardeschi, Pierpaolo Dini, Andrea Domenici, Maurizio Palmieri, Sergio Saponara. Formal Verification and Co-Simulation in the Design of a Synchronous Motor Control Algorithm. Energies, vol. 13, no. 16, pp. 4057, 2020.

[Bernardeschi 2023] Cinzia Bernardeschi, Andrea Domenici, Adriano Fagiolini, Maurizio Palmieri. Co-simulation and Formal Verification of Co-operative Drone Control With Logic-Based Specifications. Computer Journal, vol. 66, n. 2, pp. 295–317, 2023.

[Blochwitz 2011] Blochwitz Torsten and Otter, Martin and Arnold, Martin and Bausch, Constanze and Clauß, Christoph and Elmqvist, Hilding and Junghanns, Andreas and Mauss, Jakob and Monteiro, et.al.. The Functional Mockup Interface for Tool independent Exchange of Simulation Models. Proceedings of the 8th International Modelica Conference, pp. 105-114, doi=10.3384/ecp11063105

[Boeira 2018] Boeira, F., Asplund, M., and Barcellos, M. P. (2018). Mit- igating position falsification attacks in vehicular platooning. In 2018 IEEE Vehicular Networking Conference (VNC), pages 1–4. IEEE.

[Cao 2021] Cao, L. and Yin, H. (2021). A blockchain-empowered platoon communication scheme for vehicular safety applications. In 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), pp. 1–6. IEEE.

[Cho 2022] Cho Wan Joo and Kim Seongheon and Kim Yoonsoo and Moon Yong Ho, Advanced Co-Simulation Platform for UAV Simulations Under Virtual Wireless Network Environments, in IEEE Access, vol. 10, pp. 95498-95508, 2022, doi: 10.1109/ACCESS.2022.3201526

[Dadras 2018] Dadras, S., Dadras, S., and Winstead, C. (2018). Collab- orative attacks on autonomous vehicle platooning. In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), pages 464–467. IEEE.

[Dokhanchi 2018] Dokhanchi, A., Amor, H. B., Deshmukh, J. V., and Fainekos, G. (2018). Evaluating perception systems for autonomous vehicles using quality temporal logic. In Runtime Verification: 18th International Conference, RV 2018, Limassol, Cyprus, November 10–13, 2018, Proceedings 18, pages 409–416. Springer.

[Dressler 2019] F. Dressler, F. Klingler, M. Segata, and R. L. Cigno, “Cooperative driving and the tactile internet,” Proceedings of the IEEE, vol. 107, no. 2, pp. 436–446, 2019.

[ETSI 2024] ETSI GS MEC 003: “Multi-access Edge Computing (MEC); Framework and Reference Architecture” 04/2024

[Fagiolini 2019] Cinzia Bernardeschi, Adriano Fagiolini, Maurizio Palmieri, Giulio Scrima, Fabio Sofia (2019). ROS/Gazebo Based Simulation of Co-operative UAVs. In: Proc. 5th International Conference, MESAS 2018, Prague, Czech Republic, October 17–19, 2018, Revised Selected papers. Lecture Notes in Computer Science Artificial Intelligence, vol. 11472, pp. 321-334, 2019, Springer Verlag.

[Fagiolini 2020] K. Kumar and S. I. Azid and A. Fagiolini and M. Cirrincione, Erle-copter Simulation using ROS and Gazebo, IEEE 20th Mediterranean Electrotechnical Conference (MELECON), 2020.

[Farhat 2018] Nabilah Farhat and Christopher P. Ward and Roger M. Goodall and Roger Dixon, The benefits of mechatronically-guided railway vehicles: A multi-body physics simulation study, Mechatronics, Volume 51, pp. 115-126, 2018

[Fitzgerald 2015] John Fitzgerald and Carl Gamble and Larsen Peter Gorm and Kenneth Pierce and Jim Woodcock, Cyber-Physical Systems Design: Formal Foundations, Methods and Integrated Tool Chains, IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering (FormaliSE), 2015

[Gomes 2018] Gomes, C., Thule, C., Broman, D., Gorm Larsen, P., Vangheluwe, H.: Co-simulation: a survey. ACM Comput. Surv. **51**(3), 49:1– 49:33, 2018.

[Guiggiani 2023] Massimo Guiggiani, “The Science of Vehicle Dynamics” 2023

[Hotzel 2021] Hotzel Escardo Tomas and Pierce Ken and Golightly David and Palacin Roberto, Modelling Train Driver Behaviour in Railway Co-simulations, In: Cleophas, L., Massink, M. (eds) Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops. SEFM 2020. Lecture Notes in Computer Science(), vol 12524. Springer, Cham. https://doi.org/10.1007/978-3-030-67220-1_19

[Hu 2017] X. Hu, H. Wang and X. Tang, "Cyber-Physical Control for Energy-Saving Vehicle Following With Connectivity," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8578-8587, Nov. 2017

[Jawhar 2017] I. Jawhar, J. Al-Jaroodi, H. Noura and N. Mohamed, "Networking and Communication in Cyber Physical Systems," 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 2017, pp. 75-82, doi: 10.1109/ICDCSW.2017.31.

[Jia 2016] 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, pp. 263–284, Firstquarter 2016.

[Krichen, 2023] Krichen, M. (2023). Formal methods and validation techniques for ensuring automotive systems security. Information, 14(12):666.

[Larsen 2016] Peter Gorm Larsen, John Fitzgerald, Jim Woodcock, Peter Fritzson, J¨org Brauer, Christian Kleijn, Thierry Lecomte, Markus Pfeil, Ole Green, Stylianos Basagiannis, et al.. Integrated tool chain for model-based design of cyber-physical systems: the INTO-CPS project, IEEE, 2016.

[Li 2017] Li, S. E., Gao, F., Li, K., Wang, L.-Y., You, K., and Cao, D. (2017). Robust longitudinal control of multi-vehicle systems—a distributed h-infinity method. IEEE Transactions on Intelligent Transportation Systems, 19(9):2779–2788.

[Liu 2023] D. Liu, B. Besselink, S. Baldi, W. Yu and H. L. Trentelman, "On Structural and Safety Properties of Head-to-Tail String Stability in Mixed Platoons," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 6, pp. 6614-6626, June 2023

[Martinelli 2021] Martinelli, F., Mercaldo, F., Nardone, V., and Santone, A. (2021). Driver identification through formal methods. IEEE Transactions on Intelligent Transportation Systems, 23(6):5625–5637.

[Masci 2015] P. Masci, P. Oladimeji, Y. Zhang, P. Jones, P. Curzon, H. Thimbleby. PVSio-web 2.0: Joining PVS to HCI, In: Kroening, D., Păsăreanu, C. (eds) Computer Aided Verification. CAV 2015. Lecture Notes in Computer Science(), vol 9206. pp. 470–478, 2015. Springer, Cham. https://doi.org/10.1007/978-3-319-21690-4_30

[Massa 2020] F. Massa et all, “Lidar-based gnss denied localization for autonomous racing cars”, Sensors, vol. 20, no. 14, p. 3992, 2020

[MEC4AUTO 2021] https://5gaa.org/content/uploads/2021/07/5GAA_MEC4AUTO.pdf

[Mehdipour 2023] Mehdipour, N., Althoff, M., Tebbens, R. D., and Belta, C. (2023). Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges. Automatica, 152:110692, 2023

[Mirabilio 2022] M. Mirabilio, A. Iovine, E. D. Santis, M. D. D. Benedetto and G. Pola, "Mesoscopic Controller for String Stability of Platoons With Disturbances," in IEEE Transactions on Control of Network Systems, vol. 9, no. 4, pp. 1754-1766, Dec. 2022

[Mosbach 2017] Mosbach, S., Flad, M., and Hohmann, S. (2017). Coop- erative longitudinal driver assistance system based on shared control. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 1776–1781. IEEE.

[Mousavinejad 2019] Mousavinejad, E., Yang, F., Han, Q.-L., Ge, X., and Vlacic, L. (2019). Distributed cyber attacks detection and recovery mechanism for vehicle platooning. IEEE Transactions on Intelligent Transportation Systems, 21(9):3821–3834.

[Owre 1996] S. Owre, S. Rajan, J. Rushby, N. Shankar, and M. Srivas, “PVS: Combining specification, proof checking, and model checking,” in Computer-Aided Verification, R. Alur and T. Henzinger, Eds. Berlin, Germany: Springer, 1996, pp. 411–414.

[Palmieri 2019] Palmieri Maurizio and Bernardeschi Cinzia and Masci Paolo, A framework for FMI-based co-simulation of human–machine interfaces, Software and Systems Modeling, Springer-Verlag GmbH Germany, DOI:10.1007/s10270-019-00754-9, 2019.

[Palmieri 2023] Maurizio Palmieri, Christian Quadri, Adriano Fagiolini, Cinzia Bernardeschi, “Co-simulated digital twin on the network edge: A vehicle platoon”, Computer Communications, Volume 212, 2023, pp. 35-47, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2023.09.019.

[Pauca 2023] O. Pauca, M. Lazar and C. F. Caruntu, "Hybrid Cost Function Distributed MPC for Vehicle Platoons With Stability and String Stability Properties," in IEEE Control Systems Letters, vol. 7, pp. 2257-2262, 2023

[Peng 2019] Peng, C., Bonsangue, M. M., and Xu, Z. (2019). Model checking longitudinal control in vehicle platoon systems. IEEE Access, 7:112015–112025, 2019.

[Pola 2019] Pola, G. and Di Benedetto, M. D. (2019). Con- trol of cyber-physical-systems with logic specifications: A formal methods approach. Annual Reviews in Control, 47:178–192.

[Pranav 2019] Pranav Kumar Singh, Sunit Kumar Nandi, Sukumar Nandi, "A tutorial survey on vehicular communication state of the art, and future research directions", Vehicular Communications, Volume 18, 2019,100164,ISSN 2214-2096, https://doi.org/10.1016/j.vehcom.2019.100164.

[Quadri 2022] C. Quadri, V. Mancuso, M. Ajmone Marsan, and G. P. Rossi, "Edge-based platoon control," Comput. Commun., vol. 181, pp. 17–31, 2022.

[Rahman 2018] Rahman, M. S. and Abdel-Aty, M. (2018). Longitudinal safety evaluation of connected vehicles' platooning on expressways. Accident Analysis & Prevention, 117:381–391.

[Rajamani 2012] Rajesh Rajamani, "Vehicle Dynamics and Control", 2012

[Reinhart 1999] G. Reinhart and M. Weissenberger, "Multibody simulation of machine tools as mechatronic systems for optimization of motion dynamics in the design process", IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399), Atlanta, GA, USA, 1999, pp. 605-610, doi: 10.1109/AIM.1999.803237

[Richart 2023] Richart Mat'ias and Vel'azquez Felipe and Ciuffardi Federico and Visca Jorge and Baliosian Javier, CoCoSim: A Tool for Co-simulation of Mobile Cooperative Robots, In: Masci, P., Bernardeschi, C., Graziani, P., Koddenbrock, M., Palmieri, M. (eds) Software Engineering and Formal Methods. SEFM 2022 Collocated Workshops. SEFM 2022. Lecture Notes in Computer Science, vol 13765. Springer, Cham. https://doi.org/10.1007/978-3-031-26236-4_22

[Riley 2010] Riley, George F. and Henderson, Thomas R., "Modeling and Tools for Network Simulation: The ns-3 Network Simulator", The ns-3 Network Simulator. In: Wehrle, K., Güneş, M., Gross, J. (eds) Modeling and Tools for Network Simulation. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12331-3_2

[Rödönyi 2017] Rödönyi, G. (2017). An adaptive spacing policy guarantee- ing string stability in multi-brand ad hoc platoons. IEEE Transactions on Intelligent Transportation Systems, 19(6):1902–1912.

[Saponara 2022] Cinzia Bernardeschi, Pierpaolo Dini, Andrea Domenici, Ayoub Mouhagir, Maurizio Palmieri, Sergio Saponara, Tanguy Sassolas, Lilia Zaourar. Co-simulation of a Model Predictive Control System for Automotive Applications. In: SEFM 2021 Co-located Workshops, vol. 13230 LNCS, pp. 204-220, 6 - 10 December 2021Springer, Cham, 2022.

[Sepulcre 2022] 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 Transactions on Vehicular Technology 71 (1) (2022) 713–724. doi:10.1109/TVT.2021.3124708

[Singh 2018] Singh, P. K., Tabjul, G. S., Imran, M., Nandi, S. K., and Nandi, S. (2018). Impact of security attacks on cooperative driving use case: CACC platooning. In TENCON 2018-2018 IEEE Region 10 Conference, pages 0138–0143. IEEE.

[Singh 2020] Singh, P. K., Singh, R., Nandi, S. K., and Nandi, S. (2020). Integrating blockchain with cacc for trust and platoon management. Cryp-tocurrencies and Blockchain Technology Applications, pages 77–97, 2020

[Vasconcelos 2024] Vasconcelos Filho, Eˆ., Severino, R., Salgueiro dos Santos, P. M., Koubaa, A., and Tovar, E. (2024). Cooperative vehicular platooning: A multi-dimensional survey towards enhanced safety, security and validation. Cyber-Physical Systems, 10(2):123–175.

[Virdis 2019] A. Virdis, G. Nardini, and G. Stea, “A framework for MEC-enabled platooning,” in Proc. IEEE Wireless Commun. Netw. Conf. Workshop, 2019, pp. 1–6.

[Walter 2013] O. Walter et all, “Smartphone-based sensor fusion for improved vehicular navigation”, 10th Workshop on Positioning, Navigation and Communication (WPNC). IEEE, 2013, pp. 1–6, 2013

[Wang 2017] Wang, Jingtao, et al. "Platoon-based traffic control for mixed autonomous and human-driven vehicles at intersections." Transportation Research Part C: Emerging Technologies, 82 (2017): 156-172

[Wang 2023] B. Wang, J. Zheng, Q. Ren, and C. Li, “Analysis of ieee 802.11p-based intra-platoon message broadcast delay in a platoon of vehicles,” IEEE Transactions on Vehicular Technology, vol. 72, no. 10, pp. 13417–13429, 2023.

[Yao 2013] Y. Yao, L. Rao, X. Liu and X. Zhou, "Delay analysis and study of IEEE 802.11p based DSRC safety communication in a highway environment," 2013 Proceedings IEEE INFOCOM, Turin, Italy, 2013, pp. 1591-1599, doi: 10.1109/INFCOM.2013.6566955.

[Zhao 2018] Zhao, X., Chen, Y., and Zhao, H. (2018). Robust approximate constraint-following control for autonomous vehicle platoon systems. Asian Journal of Control, 20(4):1611–1623.

[Zheng 2016] Y. Zheng, S. Eben Li, J. Wang, D. Cao and K. Li, "Stability and Scalability of Homogeneous Vehicular Platoon: Study on the Influence of Information Flow Topologies," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 14-26, Jan. 2016

[Zizheng 2023] Liu Zizheng and Chu Yingguang and Li Guoyuan and Zhang Houxiang, A Co-simulation-Based System Using Vico for Marine Operation, 2023