MISSIONE 4
ISTRUZIONE
RICERCA

FORESEEN

Deliverable D2.1 – Report on co-simulation framework features and configurable parameters to generate trace dataset

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

MISSIONE 4
ISTRUZIONE
RICERCA

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 D1.3: **Report on co-simulation framework features and configurable parameters to generate trace dataset**

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

Coordinator: *University of Pisa*

Deliverable No	D2.1
WP No:	WP2
WP Leader:	RU-MI
Tasks:	T2.1 - Leader: RU-MI T2.2. - Leader: RU-PI
Due date:	M7-10
Delivery date:	September 30, 2024
Authors:	RU-MI, RU-PA, RU-PI, RU-MOL

Dissemination Level:

PU	Public	X
PP	Restricted to other program 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

CACC	Cooperative Adaptive Cruise Control
DSE	Design Space Exploration tool
FMU	Functional Mockup Unit
V2V	Vehicle to Vehicle
V2E	Vehicle to Edge

1. Introduction

The objective of this deliverable is the description of the generation process for behavioural traces of the nominal behaviour of the platoon. In particular, the co-simulation schema, the parameters setting of the design space exploration tool, and the collected data will be described. The models have been extended in order to support the co-simulation of at most 20 vehicles in the platoon. Different multi-models for V2V and V2E communication infrastructure are analysed. This part of the data gathering will be an input for WP3, which is related to the application of formal methods for identification of patterns representative of attacks starting from data traces. The co-simulation framework and the platoon multi-model used are reported into D1.1 and D1.2.

2. Co-Simulation schema

The co-simulation schema that we decided to use is shown in Figure 1.

Depending on the type of network that will be used, the schema will slightly change:

- V2V communication network. The V2V multi-model has the following FMUs: (i) one FMU for the Leader. This FMU will provide the acceleration, speed and position of the leader to any other vehicle's CACC algorithm in the platoon. (ii) the CACC algorithm FMU, one for each vehicle. This FMU will receive data from the leader, from the preceding vehicle and from the vehicle itself, in order to provide the latter with a desired acceleration. (iii) the vehicle dynamics FMU, one FMU for each vehicle. This FMU will receive the acceleration and modify its state accordingly, providing its speed, acceleration and position to the vehicle behind it. (iv) the V2V network FMU. This FMU will simply allow communication between vehicles in the platoon, modeling delay and packet loss.

- V2E communication network. The V2E multi-model is equal to the V2V described beforehand, but there will be no CACC algorithm's FMU for any vehicle. This is because the CACC is integrated in the V2E FMU, that will take the position, speed and acceleration of vehicles and provide each vehicle with the computed desired acceleration.

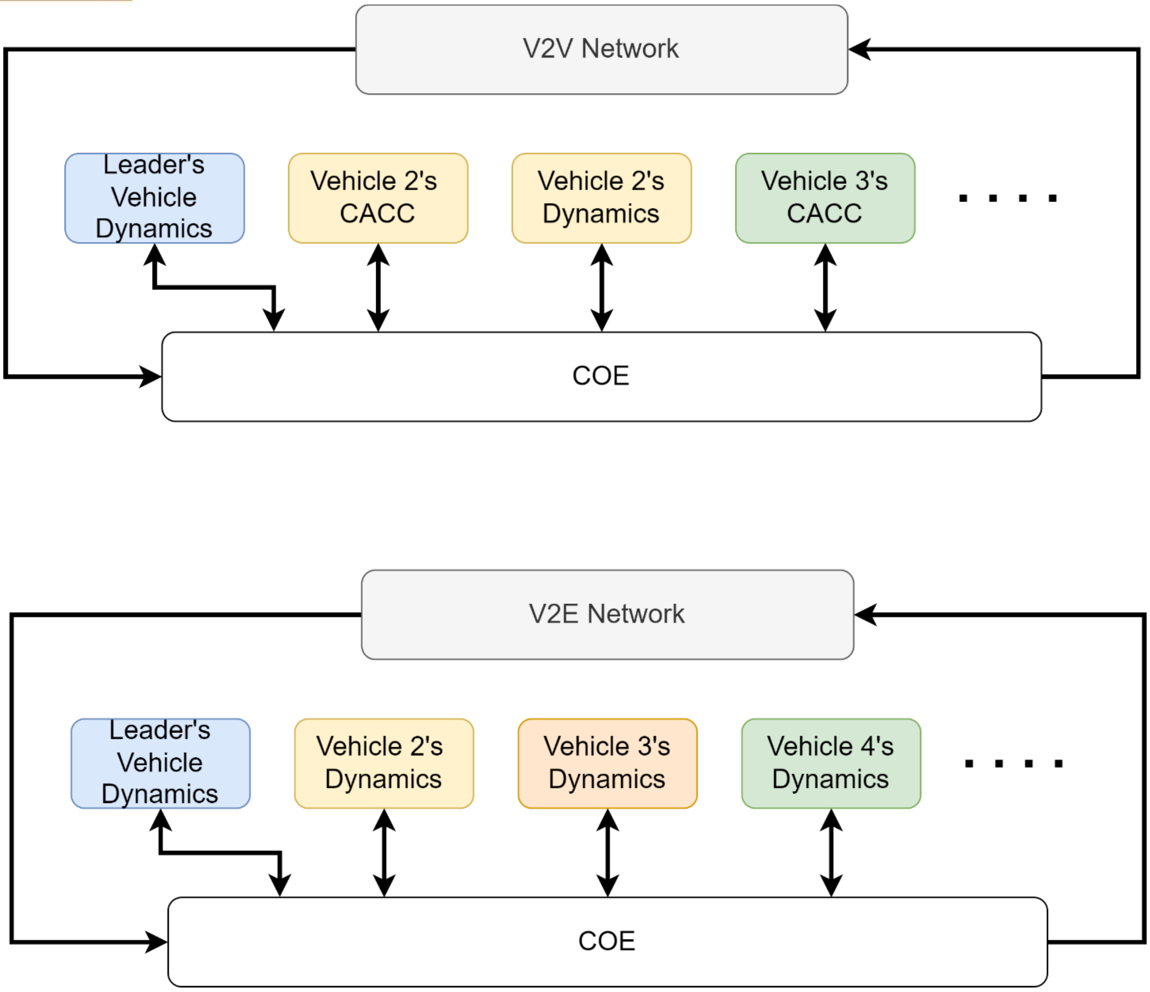

Figure 1: Co-simulation schemas: V2V and V2E network

3. From Model to FMUs

The models designed in WP1, Task T1.2, have been used to create FMUs that can be used in the INTO-CPS environment to run simulations. More precisely, before the exportation process of models as FMUs, a few adjustments have been made:

- The model of a single vehicle (the dynamics) now has a translation mechanism that allows to convert a desired acceleration, obtained by the CACC algorithm associated to the vehicle, into the torque value that will be used for the wheels. Also, a filter is provided in order to nullify the input of the CACC at the beginning of the co-simulation, to simulate the creation phase of the platoon.

- The model of both the V2V and V2E networks have been extended in order to support the co-simulation of at most 20 vehicles in the platoon. Moreover, the typical issues related to the network, i.e. delay and packet loss, are modeled in a random fashion, but they can be still reproduced by simply setting a seed before running the co-simulation.

- A special vehicle has been made starting from the model of the general vehicle's dynamics, the Leader. Its desired acceleration does not come from the CACC algorithm but from a set of pre-defined acceleration functions, created ad hoc in order to model different interesting behavioral patterns. In our case we have 3 functions:
    - Constant Speed: the leader accelerates at the beginning with an increasing acceleration, the acceleration is sets to a constant speed
    - Sine Function: the leader starts by accelerating incrementally, then after a while its desired acceleration becomes a sine function
    - Trapezoidal Function: similar to the latter, but less smooth

Each model, before being exported, has been also extended to provide a set of parameters that can be set at configuration time, before running a co-simulation. This is very useful for the next step, in which the Design Space Exploration tool [Gam2017] is applied to collect simulation data.

In particular, in order to allow the configuration of the Leader's behavior, a new parameter, **operational_mode**, has been added to the model of the Lead Car.

For what concerns the follower vehicles, to properly model the platoon creation phase we provide a period of time parameter (**vehicle_starting_time**) in which the CACC output gets ignored and other two parameters, the **initial position** and **velocity**, are set to 0.

Finally, the network admits parameters that define the **uplink** and **downlink** delay, and **packet loss** probability.

4. Simulation and Design Space Exploration

Using the INTO-CPS application [Into-cps] it is possible to manually generate a multi-model by simply connecting each output of each FMU to the associated input of another FMU. In our case, we would have a lot of input and output ports to connect, thus providing a wide range of mistakes.

For this reason, a Python script has been developed in order to automatically modify the json configuration file, used by INTO-CPS, for reading information about the multi-model and properly connecting each FMU.

Then, a few simulations were run in order to validate the system's behavior and determine a range of values of interest that will be explored during the Design Space Exploration phase, such as the shape of the acceleration function used by the leader, the seed for the network's random number generator etc.

We used the following co-simulation parameters:

- 1 Leader and 9 followers
- Co-simulation setpsize: 0,01 sec
- Co-simulation time: 125 sec
- Followers at time t=0 have acceleration equal to zero and position 0
- Each vehicle starts after 4 seconds from the preceding vehicle's starting time, and the leader starts moving at time 0
- The leader's behavioral pattern

The Leader's behavior is the following:
at the beginning it starts with an acceleration phase, this is done to ensure a proper initialization phase of the platoon. For the Trapezoidal and Sinusoidal behavior, such acceleration is set to a configurable constant value, while for the Constant behavior type, the acceleration phase takes the value from a ramp function, with a configurable slope.
After the initial phase, we have either:

1. A Sine function with configurable frequency and amplitude. Here we want high values for the amplitude and small frequency, in order to model traffic. (Sine)
2. A Sine function with very low amplitude and high frequency, to model the fact that in a real scenario it's not possible to obtain a constant speed, thus the acceleration is almost never set to 0 constantly. (Constant)
3. A train of impulses, where we alternate acceleration and deceleration phases, separated by periods of time in which the acceleration is a low amplitude sine function around 0 (Trapezoidal)

Figure 2 shows the position of the vehicles in a co-simulation run. Figure 3 shows the acceleration of vehicles. In particular, in Figure 3 we can observe that in the first part of the simulation (40-50 sec), in order to model the platoon creation phase, each vehicle ignores the CACC's desired acceleration. After the initial set-up, the figure shows that followers adjust their acceleration to the one of the Leader.

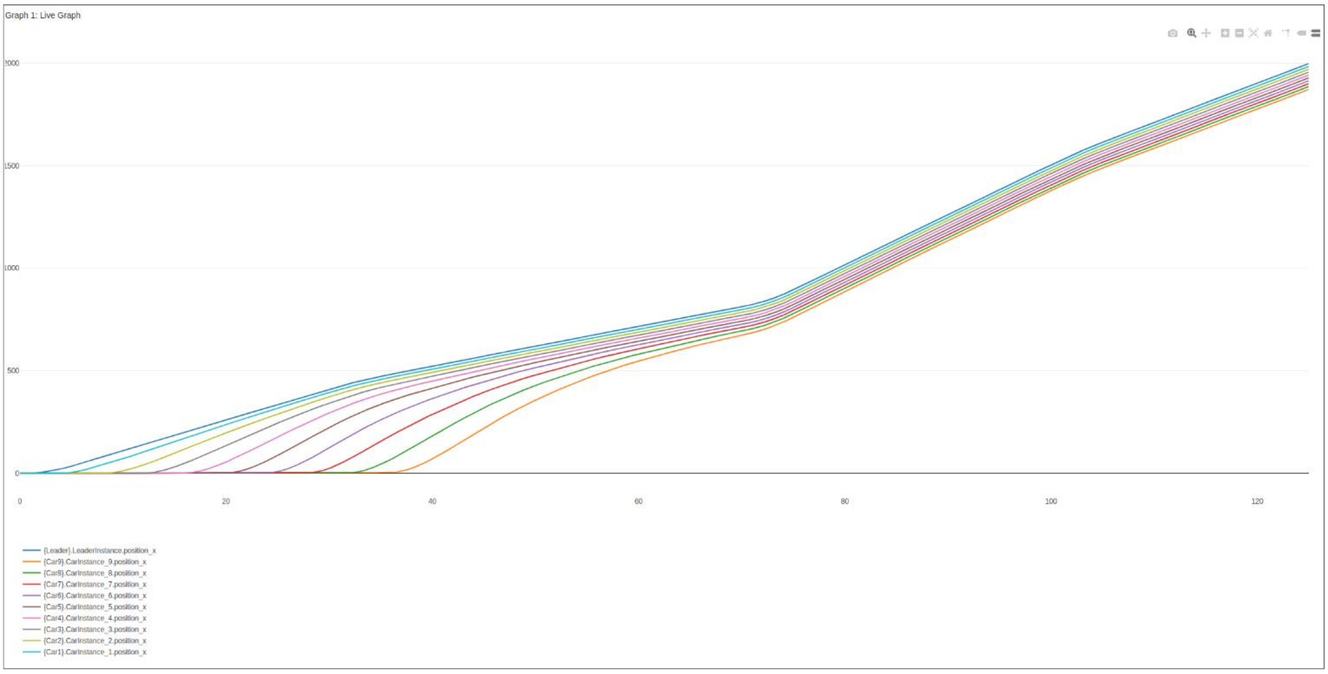

Figure 2: Co-simulation of the platoon: vehicle's position

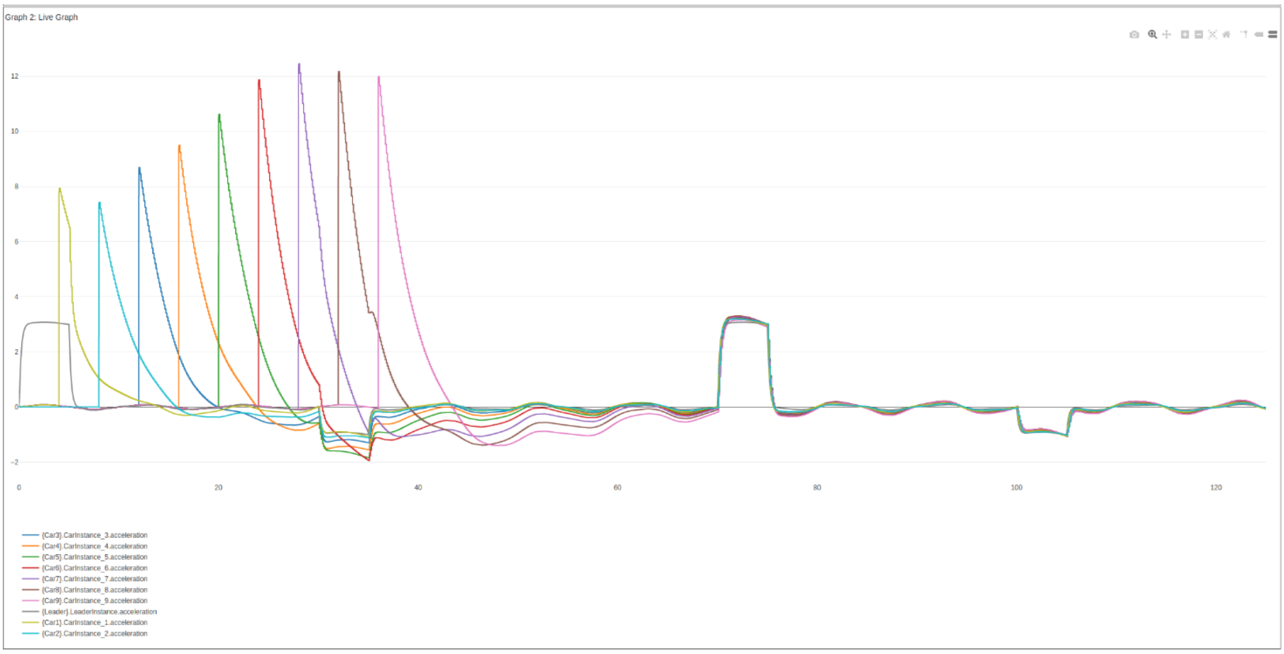

Figure 3: Co-simulation of the platoon: vehicle's acceleration

5. Data collection

A set of automatically run co-simulations has been made, using configurations of parameters obtained using the exhaustive search algorithm on every value of the list of values for each parameter of the multi-model.

Data collection is organized for run of simulation: one .csv file for each simulation is generated. The .csv file contains data of the co-simulation step by step.

Data collection parameters: co-simulation stepsize: 0.01 sec and co-simulation duration: 125 sec.

5.1 DSE configuration

Parameters that are objective of the DSE configuration:

- Leader acceleration: the leader behavioral pattern; for each pattern the configuration parameters analyzed in the previous chapter
    - Frequency of both the sine functions
    - Amplitude of both sine functions
    - Slope of the ramp function
    - Period of acceleration and deceleration in the trapezoidal train
    - Amplitude of the acceleration
    - Amplitude of the deceleration
- Network delay: a seed is used to generate random delay and packet loss in real scenarios. Such delay and packet loss rate are tolerated by the CACC law.
  Other configurable parameters could be , for example, the length of each vehicle in the platoon.

In the case of V2E network, the configuration file of the DSE is the following:

```
{
  "algorithm": {
    "type": "exhaustive"
  },
  "objectiveConstraints": [],
  "objectiveDefinitions": {
    "externalScripts": {},
    "internalFunctions": {}
  },
  "parameterConstraints": [],
  "parameters": {
 "{Leader}.LeaderInstance.initial_position": 0,
 "{Leader}.LeaderInstance.initial_velocity": 0,
"{Leader}.LeaderInstance.operational_mode": [ 0,1,2 ],
"{Network}.NetworkInstance.platoon_0_0_length": 4,
"{Network}.NetworkInstance.platoon_0_1_length": 4,
"{Network}.NetworkInstance.platoon_0_2_length": 4,
"{Network}.NetworkInstance.platoon_size": 10,
"{Network}.NetworkInstance.platoon_0_3_length": 4,
 "{Network}.NetworkInstance.platoon_0_4_length": 4,
 "{Network}.NetworkInstance.platoon_0_5_length": 4,
"{Network}.NetworkInstance.platoon_0_6_length": 4,
"{Network}.NetworkInstance.platoon_0_7_length": 4,
"{Network}.NetworkInstance.platoon_0_8_length": 4,
 "{Network}.NetworkInstance.platoon_0_9_length": 4,
"{Network}.NetworkInstance.network_downlink_delay": [  0,0.1 ],
 "{Network}.NetworkInstance.network_uplink_delay": [ 0,0.1],
"{Car1}.CarInstance_1.initial_position": 0,
"{Car1}.CarInstance_1.initial_velocity": 0,
"{Car1}.CarInstance_1.vehicle_starting_time": 8
},
..... similarly for all the other vehicles

  "ranking": {
    "pareto": {}
  },
  "scenarios": []
}
```


In this case, the network sends accdes value to each follower. In the file, Platoon01 denotes vehicle 1 of the platoon.

5.2 Parameters in the .csv file

The output of this phase is a set of .csv file which record the results of the simulation. The .csv store the following data types:

- Time_step,
- For each vehicle:
  - The speed
  - The position
  - The acceleration
  - The desired acceleration sent by the CACC
  - Whether the vehicle is under attack or not, this is used to validate the results during the development of formal models and checking properties.

The .csv file is organized as follows:

time
{Network}.NetworkInstance.platoon_0_1_des_acc
{Leader}.LeaderInstance.position_x
{Leader}.LeaderInstance.speed
{Leader}.LeaderInstance.acceleration
{Leader}.LeaderInstance.attacked
{Car1}.CarInstance_1.position_x
{Car1}.CarInstance_1.speed
{Car1}.CarInstance_1.acceleration
{Car1}.CarInstance_1.attacked
..... similarly for all the other vehicles

The data set is available through the remote Foreseen machine of the FoReLab at RU-PI.
Part of the data will also be uploaded on the website of the project.

6. Roadmap for further work

In the next tasks of WP2, the framework features and configurable parameters defined in this deliverable will be extended to gather execution traces of the platoon in case of attacks. Moreover, the generated data set will be input to WP3, to build formal models for threats identification from execution traces of the cyber-physical system.

7. Bibliography

[Into-cps] INTO-CPS Online Documents.
[Online] https://intocps-Association.readthedocs.io/en/latest

[Gam2017] Gamble, C., “DSE in the INTO-CPS Platform”. INTO-CPS Deliverable, Aarhus Univ., Denmark, Tech. Rep. D5.3e