Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

A simulation-based testing and validation framework for ADAS development

Tong Duy Son, Jacob Hubrechts, Lisa Awatsu, Ajinkya Bhave,
Herman Van der Auweraer

Siemens Industry Software NV (SISW), Interleuvenlaan 68, 3001 Leuven, Belgium

Abstract

This paper presents a virtual validation and testing framework for ADAS (Advanced Driver Assistant System) planning and control development based on a co-simulation platform of vehicle dynamics and traffic environment tools. One of the main challenges in ADAS development is validating the planning and control algorithms in a closed-loop fashion, where both vehicle dynamics characteristics and a wide variety of traffic scenarios are taken into account. The designs should also guarantee optimal performance toward precise trajectory tracking, and time/fuel optimality with respect to various constraints. This work focuses on simulation-based approaches to frontload control design verification during the early phases of ADAS development involving two software: LMS Imagine.Lab Amesim and PreScan. The requirements for an interface that help to facilitate the co-simulation development are studied. The approach is demonstrated with three different use cases: adaptive cruise control, green wave technology, and autonomous parking.

Keywords: ADAS, autonomous vehicles, control, co-simulation.

1. Introduction

ADAS (Advanced Driver Assistant System) and AV (Autonomous Vehicles) technologies that aim for more safety, comfort, and environmentally friendly vehicles, have been growing rapidly in automotive industry recently. These systems go beyond vehicle-centric assistant applications such as automatic emergency braking (AEB) or traction control by integrating traffic and environment information. This involves collaborative research and development activities from different fields such as mechanical, electrical engineering, control, communication, and computer science. In this work, we present a co-simulation based framework and technologies to frontload testing, verification and validation of ADAS/AV systems during the early phases of development. The simulation toolset takes into account vehicle dynamics, traffic environment, V2V/V2X technologies, sensor models, planning and control algorithms, and will be demonstrated via three ADAS applications: adaptive cruise control, green wave technology, and autonomous parking.

It is getting more recognized in the autonomous driving industry that simulation based testing is an efficient method to validate ADAS/AV functionalities. The traffic environment has a wide variety of road types and conditions, vehicles, pedestrians, cyclists, obstacles, and weather, where each scenario has multiple combination of parameters. The number of scenarios grows exponentially with the number of parameters and can explode up to millions. Typically, in order to achieve a reliability validation using real car testing, the car needs to drive millions to billions of kilometres which is extremely time consuming and expensive [Kalra and Paddock (2016)]. Furthermore, all scenarios cannot be produced and reproduced easily in real life. It is worth noting that real road testing is valid for a specific mechanical, electrical and software configuration. If needed to adapt to a new configuration or software update, the test has to be conducted again. This is why the major part of the ADAS functionalities should be validated through massive simulation, and hence Model in the Loop (MiL) and Software in the Loop (SiL) verifications will have to play a bigger role in the development process.

In this paper, we focus on the development of the planning and control components of the ADAS/AV systems using the proposed co-simulation framework. The planning algorithm decides a desired trajectory that the vehicle need to follow, and the controller steers the vehicle to that trajectory. Developing and verifying these algorithms are challenging tasks due to several reasons. First, they need to guarantee safety with respect to a wide variety of traffic scenarios, while taking into account various vehicle dynamics constraints such as non-holonomic property, and acceleration and steering constraints. Second, the car should drive in a similar manner as human driving, that is, not only in a safe and legal fashion but also smooth and comfortable. This is challenging especially when driving in urban areas or at high speed on highway. Third, as one of the main drivers of autonomous driving technology, customers require that the autonomous vehicles will have to be more environmentally friendly and fuel efficient . The increasing government requirements on fuel economy and pollution are also impacting the control specifications. Finally, it is essential to consider robustness in the planning and control algorithms to various uncertainty sources, from both vehicle modelling and environment sides. Models for physical systems always have inaccuracies (due to non-perfect measurements, production errors, or aging problems) and missed dynamics are inevitable because they are too complex to be taken into account. From the environment side, sensing, mapping, and localization algorithms are inherently non-deterministic due to measurement noise and uncertain environment conditions. Considering all these challenging aspects, it is desirable to verify the planning and control developments in a simulation-based framework, where all the physical and environment factors are included.

As discussed, ADAS/AV system development incorporates multidisciplinary knowledge from different components. A co-simulation framework that can combine these components and frontload the design development in the early stage of functional development is crucial. This framework should be sufficiently flexible in order to adapt shared learning from industry and academic collaborators, and at the same time protect their key intellectual property rights. On the other side, researchers/engineers working on one component usually have some general assumptions about the other components. For instance, while the planner designs a reference trajectory that satisfies a certain set of constraints, it normally assumes accurate sensing, simple vehicle model, and that the tracking controller will be able to follow this trajectory accurately. These assumptions are certainly not always guaranteed with all vehicle dynamics and sensing configurations. In this sense, the co-simulation toolset helps to verify and improve the design development in a deterministic and structured manner.

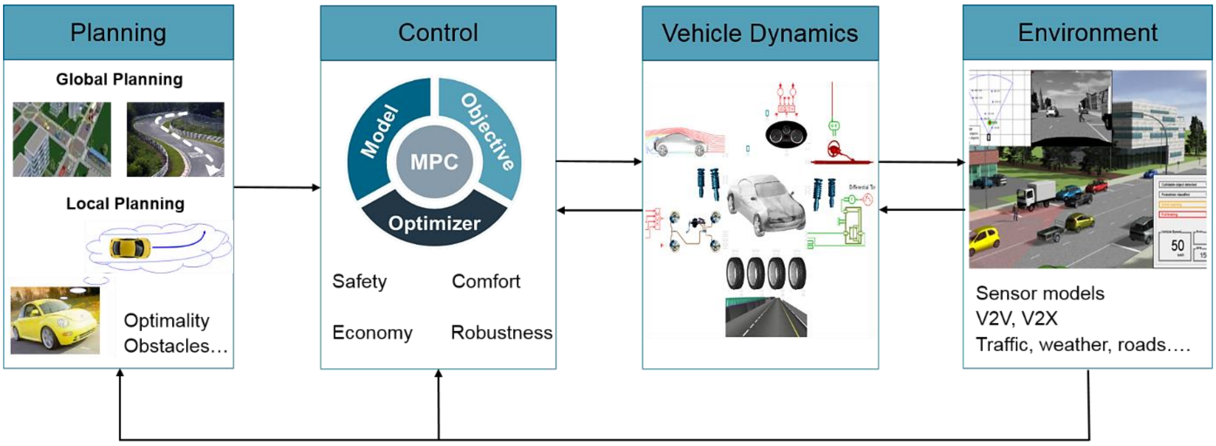

Fig. 1 ADAS/AV Co-simulation structure

The proposed ADAS/AV testing framework is based on the co-simulation of two software: LMS Imagine. Lab Amesim and PreScan. Amesim provides an integrated simulation platform to accurately predict the multidisciplinary performance of intelligent systems in general and vehicle dynamics specifically. It offers plant modelling and simulation capabilities to connect to controls design, helping to assess and validate control strategies. PreScan is a physics-based simulation platform that can simulate complex traffic scenarios and ADAS/AV sensor technologies such as radar, lidar, ultrasound, and camera. PreScan can also design and evaluate vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication applications. The combination of these two simulation frameworks, one focused on the vehicle dynamics and one focused on the environment outside the vehicle, yields a basis for the virtual validation and verification of autonomous driving functionalities.

The co-simulation structure is demonstrated in Fig. 1. This toolset provides an open, powerful, and user-friendly platform for system modelling and analysis, and especially scripting capabilities to support application programming in higher abstraction level languages (Matlab, Python,…). Hence it can easily integrate with planning and control algorithms for test frontloading. In this work, the ADAS development using the proposed framework is divided into three main steps. First, the algorithms are developed with a ground truth data of the ADAS application and only Amesim vehicle dynamics. Further works on modelling, design of experiment, and optimization on vehicle-centric configurations (chassis, powertrain, braking models…) can be conducted in this stage. After that, the designed algorithms are validated with PreScan traffic and sensor simulation models. The scenarios and performance requirements follow mainly industrial standards, realistic or corner cases. The study of how to build the co-simulation configuration and variable exchanges (vehicle, sensor, environment, and controller model variables) are investigated for each specific ADAS application. Finally, the results are evaluated with respect to the system requirements, i.e. on safety, comfort, natural driving, and ecology. The co-simulation setup enables to test all model aspects, validate sensor precision and improve visualisation. Compared to real prototype testing, the high repeatability, low marginal cost, and ease of changing model parameters are the major benefits of simulations.

It is worth stressing that the multi-physics modelling software Amesim has been investigated extensively from both industry and academic sides for Model-Based Systems Engineering (MBSE) development. For example, Wissel et al. (2016) applied Amesim for their powertrains system development together with model predictive control (MPC) technique, and Vanhuyse et al. (2016) designed nonlinear MPC for Amesim hybrid vehicles. However, these works haven't considered the benefits of Amesim for ADAS applications, and hence lack of integration to environment and sensor modelling software like PreScan.

The paper is organized as follows. Section 2 provides more details on the technologies of the software and algorithms used in the co-simulation framework: Amesim, PreScan, planning and control designs, and the co-simulation setup. Demonstrations on the adaptive cruise control (ACC), green wave technology, and autonomous parking are given in Section 3. Finally, Section 4 concludes this paper.

2. Technology

In this section, the main technologies which are used in the co-simulation are presented.

2.1. Vehicle dynamics

LMS Amesim is a multi-physics simulation platform that provides libraries of different physical domains such as fluid, thermal, mechanical, electromechanical and powertrain. In this work, it will be used to simulate a vehicle and its dynamics for the purposes of testing ADAS/AV planning and control algorithms. The fact that this software handles various different physical domains makes it a very practical integrated solution in the automotive industry. The extensive libraries can speed up modeling work substantially. For example, when designing a headlight-angle controller, engine dynamics are not very important. In that case a simple pre-made engine block can be used. This approach ensures no time is lost on modeling non-crucial components and little computing time is consumed by non-crucial components. By connecting blocks, a complex and realistic model can easily be constructed for system level simulations; on the other hand, very detailed simulations of single components can be made as well. Fig. 2 shows the model of a complete vehicle chassis that is used in our use cases, including powertrain, braking, suspension, steering components.

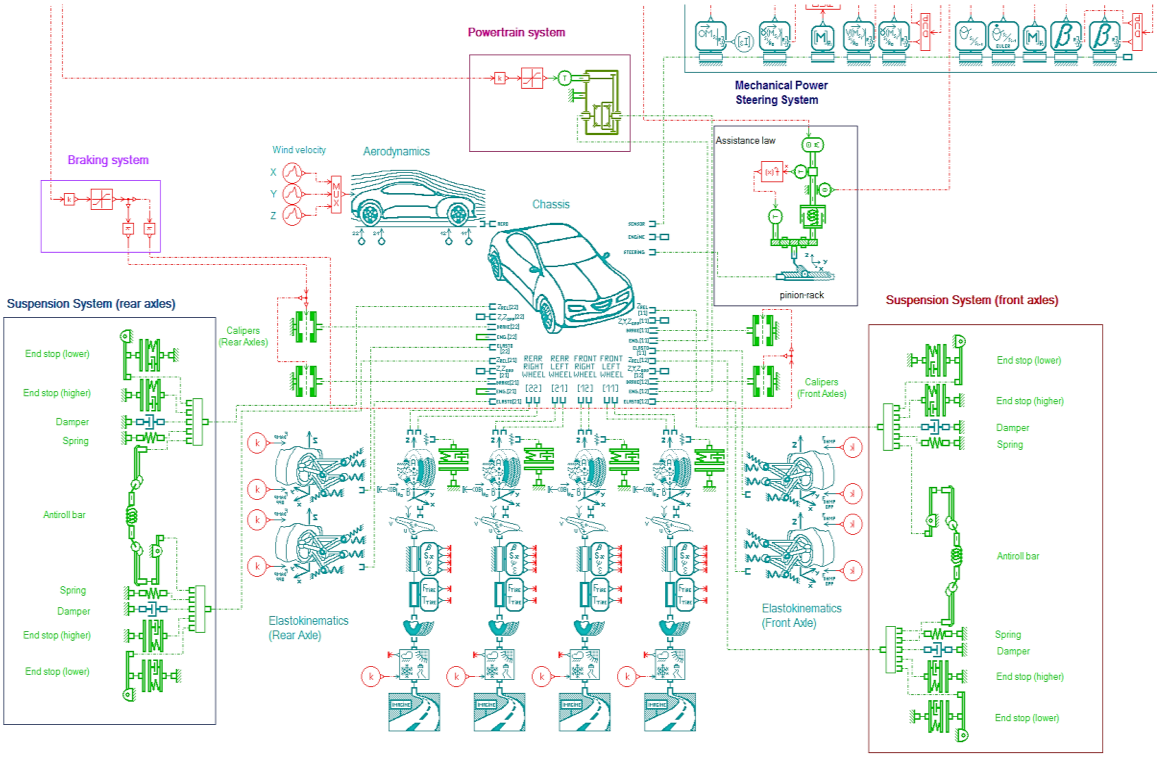

Fig. 2 Amesim vehicle dynamics simulation of the vehicle chassis

2.2. Environment

PreScan is used in the automotive industry to aid in the development of ADAS systems. It has an intuitive graphical user interface (GUI) with extensive environment modeling capabilities, allowing users to build and modify traffic scenarios, including road sections, infrastructure components (buildings, traffic signs), actors (cars, trucks, cyclists, pedestrians), weather conditions (rain, snow, fog) and light sources (the sun, headlights, lamp posts). The vehicle models can also be equipped with ADAS sensors that sense the virtual environment such as radar, lidar, camera, ultrasonic, GPS, and antennas. ADAS systems are designed to reduce the driver workload by interpreting the environment and acting accordingly to that. The simulation of such a system depends on how realistically the environment and sensors can be simulated. That is where PreScan is powerful. Furthermore, PreScan has advanced visualization capabilities. The benefit of good visualization is twofold. On one hand, it helps the engineer developing new ADAS system functionality. On the other hand, good visualizations are useful to communicate work progress to higher management and customers. A screen shot from a PreScan visualization video is given on the environment block in Fig. 1.

2.3. Planning

The planning designs depend on specific ADAS applications. For example, in ACC, the planner generates reference velocity and steering profiles in order to obtain safe distance to the car driving in front. This function is usually used on highway at high speed, thus the computation must be sufficiently fast, and vehicle dynamics analyses are important. The green wave technology usually happens on a straight road/lane, where the car receives a reference velocity from the traffic light infrastructure via V2X communication. The planning algorithm is mainly implemented from the infrastructure side and less important from the vehicle side. In autonomous parking, the car drives at low speed; consequently, the planning computation time is less critical than driving on highway. However, the parking area often have limited space and it is expected there are more types of obstacles (other cars, pedestrians, objects…) from different directions. In some cases, when changing gears (forward/backward) and directions are needed, the task becomes harder. The planner needs to consider all these factors into the design. Generally, the planning stage in our use cases is divided into two main steps:

- Global Planning: Planning the route (or way-points) at the high level. This is applied in the parking use case, where the global planner generates roughly the path to follow from the initial point to the goal parking lot, for example, based on the shortest route.
- Local Planning: Given a reference path (or way-points) from the global planner, the vehicle will plan its own detailed trajectory, including timing profile. But it is not necessary for the vehicle to follow the path very precisely as the following requirements are more critical: 1. Vehicle constraints such as non-holonomic constraints (i.e. the vehicle cannot move to any configuration), steering, velocity, and acceleration constraints. 2. Avoiding obstacles: for example, an unexpected obstacle (pedestrian, other moving cars) was encountered during the execution of the pre-computed path. Local planning is thus an online planning stage. 3. Optimality with respect to a given criteria such as timing or fuel consumption.

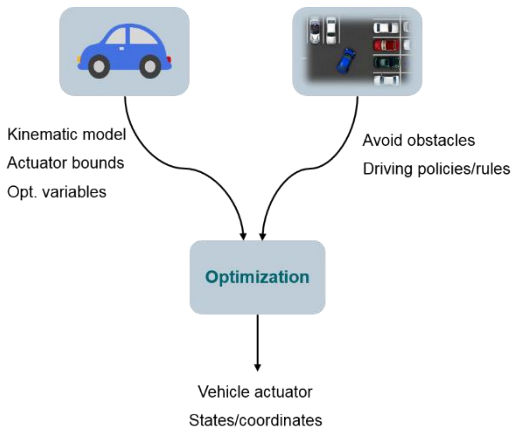

Fig. 3 Local planning structure

Fig. 3 presents the local planning structure that is used in the autonomous parking application. Note that there could have been another layer of planning in other ADAS systems, i.e. traffic planner to decide change lane, or over taking but it is not the focus of our use cases.

2.4. Tracking control

The controller is designed to control the vehicle along the computed trajectory. The controller is crucial because of uncertainty reasons, typically plant model mismatch and disturbances to the vehicle. A PID control is developed as a comparison baseline. PID controller has simple implementation and well understood characteristics. The fact

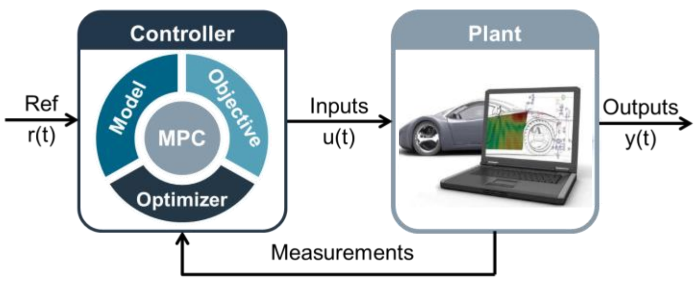

Fig. 4 Model predictive control (MPC) structure

that the PID is mathematically simple and a black-box type control (no modelling required) makes it a practical tool and it is widely used in industrial applications.

Our main focus on control design is model predictive control (MPC) development that can deliver optimal control over a wide range of operation conditions. MPC design relies on solving a model-based optimization problem, and has two main advantage over PID control: can deal efficiently with multi-input multi-output systems and vehicle system constraints. Recently, MPC is being actively investigated in automotive industry, where the algorithm, implementation, and tuning are found to be quite efficient.

2.5 Co-simulation structure

In this subsection, the presented technologies are combined in the co-simulation structure.

2.5.1 Ground truth design

The first tuning step is performed in the ground truth simulation environment of Amesim. In order to be able to tune the controllers for optimal performance, objective performance measures are needed, based on which the tuning process can take place. In this stage, incorporation of sensor dynamics and environment influences on the measurement signals are not yet of any concern. The Amesim solver runs the vehicle model, where all Amesim-related functionalities can be used and the model variables are readily accessible. The main goal now is to get the controller dynamics set up correctly so that the controller performs well in ideal circumstances. The controller can be developed from Amesim itself or from other environments (i.e. Matlab, Python).

2.5.2 Co-simulation setup

PreScan is then incorporated in order to simulate environment and sensors. This allows testing and validating planning and control with more relevant scenarios, steering away from the ground truth assumptions. The decomposition of the co-simulation setup is not straightforward as some parts can be moved from one software package to another and vice versa. This should be taken into account for the specific ADAS application and the purpose of testing. Clearly, different possible configurations can cause confusion when collaborating as a team. To facilitate the work flow, an interface that connects different models has to be agreed upon before starting the project. This interface defines what variables are exchanged between the models and in which manner.

As an example, we present the model decompositions and variable exchange interface of the ACC use case. An environment is simulated in PreScan, and the sensors that are mounted on the ego vehicle sense the environment. The output signals of those sensors are led from PreScan to the ACC controller in Simulink. The current acceleration and yaw rate are also used by the controller, those signals come from the Amesim model. The controller calculates throttle and brake signals. Those signals are then fed into the Amesim vehicle model. A path follower in PreScan also calculates a suitable steering signal that keeps the vehicle on a predefined path. To be able to do this, the path follower needs the yaw angle from the vehicle model. The steering signal is fed through Simulink into the Amesim model. Moreover, for each of the four tires, a road friction coefficient value (based on the position of the car and the ground material) is sent through Simulink to the Amesim model. Given those inputs (tire friction, steering angle and throttle/brake signal) the states (position and velocity) of the vehicle can be calculated through Amesim solver. The new vehicle states are then used to update the states of the vehicle model in PreScan. Accordingly, environment sensing is updated, and the circle continues.

3. Use Cases

Three different use cases are demonstrated: ACC, green wave technology, and autonomous parking, which are typically at high speed, middle range speed, and low speed, respectively. They use three different co-simulation decomposition/interface setups and planning/control design developments. The V2V (vehicle-to-vehicle) and V2X (vehicle-to-everything**)** technologies are also included. The development of an ACC system is first discussed in details to gain insights of the proposed framework, and then the other two use cases will be briefly presented for the sake of saving space.

3.1. Adaptive Cruise Control (ACC)

The ACC development relies on a key performance index (KPI). This is a number or set of numbers that describes the performance of a system, and can also be used to compare the performance of different ACC systems in an efficient and practical way. The basis of the KPI calculation is often a benchmark that is set by industry standards. Similar ACC design development and validation can also be found in the recent work in Beglerovic et al. (2017). In this research, we consider the following four characteristics to describe an ACC controller performance.

- Safety: Describes how the ACC controller avoids and handles dangerous situations. Collisions should be avoided at all times. A quantifiable parameter that describes how a controller avoids getting into collision-prone situations is the Time To Collision (TTC), measured in seconds.

- Comfort: Describes how gentle the controller is, to ensure a comfortable ride for passengers. To achieve desirable comfort levels, the ISO standard sets limits on the tolerated acceleration and deceleration levels. The result of this integration is a KPI number that represents comfort performance of the controller.

- Natural driving: Considers how "human" the controller is behaving in traffic situations. The maintained following distance is counted in this case. This is determining for the mental comfort of the passengers. If an ACC system behaves different from how an average person would act, it would give an unpleasant feeling to the passengers and the driver would feel a constant urge to retake control over the car. Naturally, drivers keep a velocity-dependent following distance which can be expressed in seconds, called the head time.

- Ecology: How fuel economy and pollution are affected by the controller, and relates to the torque demand. It is known that high torque demands will result in higher engine emissions. Diesel engines have an increase in particulate matter emissions and gasoline engines suffer from increased hydrocarbon and NOx emissions.

3.1.1. Ground truth testing

The above mentioned methods allow to calculate four different KPIs: safety, comfort, natural behavior and ecology. The next step is to create scenarios in which the controller can be tested. The scenarios are simulated in a ground truth environment. That means, for example, that the controller disposes of input variables that are always accessible and fully accurate. It can be comparable to the situation of perfect sensors that have infinite resolution, infinite sample rate and zero noise. This ground truth environment is created in a co-simulation with Amesim and Simulink. The vehicle model runs in Amesim, the controller model runs in Simulink. The exact signals like e.g. relative velocity, following distance, velocity, and acceleration can be extracted from the Amesim model during simulation and fed into the controller. To make the actual decision, one has to take into account the importance attached to different driving characteristics like safety, comfort, natural driving behavior and fuel economy. Based on the relative importance, a weight can be assigned to each of the four KPI numbers of each controller. The controller tuning relies on two steps. The first step involves finding out what parameters can be configured in order to tune the controller for optimal performance. For example, in the PID controller it is straightforward that the configurable parameters are P, I and D gains. In the MPC controller, it is characterized by the set of weights used in the objective function. The second step finds the optimal parameter configuration that delivers better results than other configurations. This can be done by testing stochastically or exhaustively with different parameter sets; there are also some design of experiments tools can help with that.

3.1.2. Co-simulation Amesim-PreScan testing

It is clear that the controller developed in the ground truth environment cannot be used to process realistic sensor signals. The controller developed in the ground truth environment doesn’t need to distinguish between several possible targets. Only the data of one vehicle just in front of the ego vehicle was fed into the controller. Realistic sensors on the other hand sense multiple targets at the same time and the controller has to decide first what target is in the lane in front of the ego vehicle. In order to create a working controller, the ground-truth controller has to be expanded with an extra module: the target selection module. The controller is then adapted and ready to function with realistic sensor output. The remaining works are modeling the sensors and the environment on PreScan.

Three scenarios will be built in the PreScan software, according to the tests prescribed in the ISO standards. The first scenario will test the discrimination capabilities of the developed ACC systems. The second scenario will test the re-targeting capabilities. Re-targeting means that the controller can autonomously judge, based on sensor output, what vehicle drives in front of it and adapt its judgment when the current traffic situation changes due to, for example, a cut-in or cut-out event. The third scenario will test the curve capabilities of the controller. All those tests are performed on dry asphalt in clear weather conditions with a visibility of 1000 meter and a temperature between −20 to 40 degrees Celsius.

- Discrimination capabilities: This test measures the capability of the ACC controller to discriminate between two adjacent vehicles and to decide which one is in the same lane as the ego vehicle. The test is performed on a straight road, prescribed by ISO 15622. Vehicle 1 (ego vehicle) and vehicle 2 drive in the same lane. The ego vehicle follows in steady state following mode with the maximum allowable time gap. Vehicle 2 then accelerates, vehicle 3 stays at a constant velocity. The ACC controller interprets the situation and controls the longitudinal behavior of the ego vehicle. The test is successful if the ego vehicle correctly accelerates to follow vehicle 2 and passes vehicle 3.

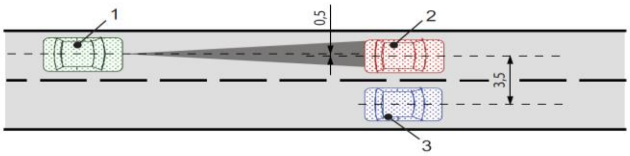

Fig. 5 Discrimination capabilities test [ISO15622]

- Retargeting capabilities: This test measures the capability of the ACC controller to react to a cut-out of the preceding vehicle and to retarget accordingly. As prescribed by ISO 22178, this test is performed on a straight road. The ego vehicle is controlled by the ACC controller and follows the preceding vehicle in steady state. The preceding vehicle changes to the right lane because there is a much slower vehicle in front of it. The test is successful if the ego vehicle decelerates and starts driving behind the slow vehicle.

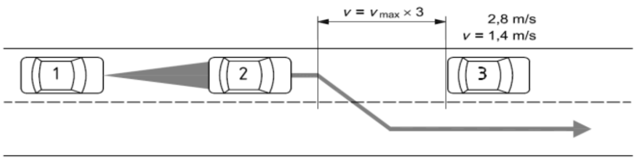

Fig. 6 Retargeting test [ISO22178]

- Curve capabilities: This test measures the capability of the ACC controller to perform steady longitudinal control while driving through curves. As specified in the ISO 15622 standard, in this case, a curve of radius 125 m has to be selected. The preceding vehicle and the ego vehicle drive in steady state in the same lane. After 10 seconds, the preceding vehicle slows down. The test is successful if the ego vehicle starts to decelerate before the time gap becomes smaller than 2/3 of the selected time gap.

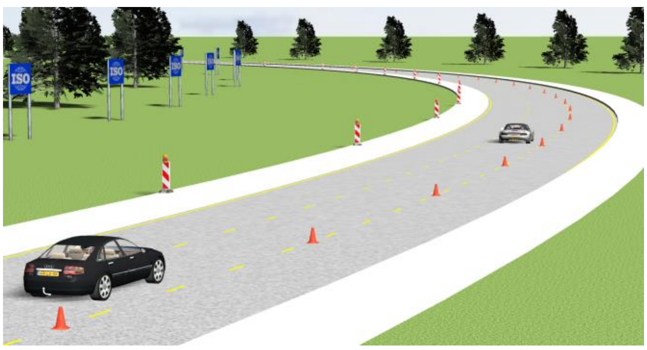

Fig. 7 ACC use case: object selection in a curve [ISO 15622]

3.2. Green Wave Technology

The main idea of the green wave technology is that the car can let the driver know what speed to maintain to get through the continuous green lights or when the traffic light will turn green by transmitting the information between the vehicle and the infrastructure. The driver can go through the traffic light without stopping at the red light. The advantages include reducing traffic jam, fuel consumptions and stress on drivers. Fig. 8 shows the PreScan demonstration of this ADAS application. The co-simulation setup and control development is similar to ACC case.

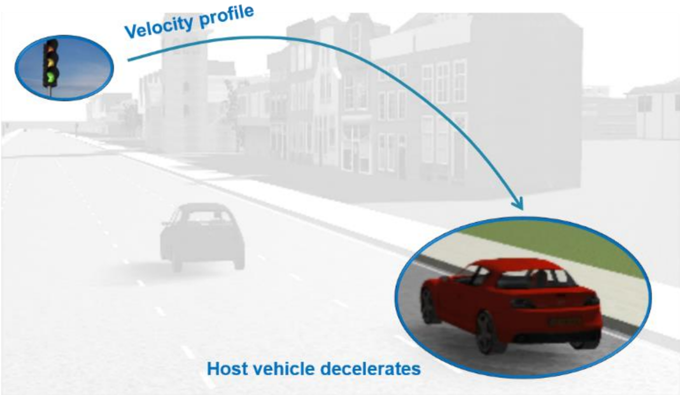

Fig. 8 Green wave technology

3.3. Autonomous Valet Parking

Valet parking is a functional extension of the parking assistance and is estimated to be one of the first commercially available fully automated driving functions since driving happens with low speed and the traffic environment can be supervised. When activated by the user, the function together with a Parking Area Management (PAM) system will fully take over control of the vehicle and the vehicle is driven through the parking area until it comes to a designated parking position. In this application, there are different planning and control levels are required:

- Global planning: Planning the path (or way-points) from the initial handover zone to the goal parking bay. This task is generated from the Parking Area Management (PAM) with a parking map.
- Local planning and control: Given the reference path from the global planning, the vehicle will plan its own trajectory to follow the defined path or way-points using its sensors and perceptions. The planner should deal with uncertainties from the parking area (pedestrians, other cars entering or leaving their parking lots).

Fig. 9 Valet parking scenario

- Maneuvering planning and control: This component considers vehicle maneuvering to the parking bay. The approaches and requirements are mostly similar to the local planning stage, but require more attention due to tight space and more obstacles (parked cars). In addition, in some cases, multiple movements with change of direction (or gear shift) of forward/backward is required. Fig. 9 shows our valet parking scenario in PreScan.

Unlike ACC and green wave technology use cases, the planning and control algorithms developments of valet parking are mainly developed in Matlab and Python, for both velocity and steering planning and control.

4. Conclusions

The co-simulation testing framework of Amesim and PreScan delivers benefits in multiple ways. First, the complete planning and controller developments can be tested and validated with high fidelity vehicle dynamics and real world environments with realistic sensors. The second benefit is that the standard test scenarios, or any other corner case scenarios, with more focus on interactions with the environment, can be used for testing. This is very valuable since it drastically reduce the amount of real prototype tests that have to be performed. In addition, the repeatability of those tests is much higher than the tests performed in a real prototype: other traffic and weather conditions are under full control of the control engineer. Another benefit is the great visualization possibilities that PreScan offers which is practical for the engineers as well as for customer contact.

On the other hand, setting up a co-simulation can be time consuming: with the user's guide of both Amesim and PreScan in hand, the co-simulation was set up step by step. A positive point is that both manuals are very exhaustive and clear. Once everything worked from the software side, making the models truly work together takes more time. For example, the extra module added to the controller for the selection of the obstacle vehicle makes use of the yaw rate and the vehicle model had to be adapted so that it could provide that yaw rate to the controller. Another example is the transfer of position and velocity vectors: depending on the coordinate system in which they are expressed, a transformation might be necessary before they can be transferred between Amesim and PreScan. The detailed co-simulation setups are demonstrated successfully via three different ADAS applications: adaptive cruise control (ACC), green wave technology, and autonomous valet parking.

Acknowledgements

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under grant agreement No 692455. This joint undertaking receives support from the European Union's HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway. This work was also supported by the Flanders Make ICON EMDAS (Environmental Modelling for automated Driving and Active Safety) project.

5. References

Kalra, N., Paddock, S., 2016. Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?. Santa Monica, CA: RAND Corporation, 2016.

Wissel, D., Thomas, V., Lansky, L., 2016. Linking model predictive control (MPC) and system simulation tools to support automotive system architecture choices, 8th European Congress on Embedded Real Time Software and Systems.

Vanhuyse, J., De Bruyne, S., Nicolai, M., Atarashi, D., Van Der Auweraer, H., Desmet, W., 2016. Nonlinear MPC design using Amesim models. TMCE 2016 Symposium. Aix-en-Provence, France.

Beglerovic, H., Ravi, A., Wikström, N., Koegeler, H., Leitner, A., Holzinger, J., 2017. Model-based safety validation of the automated driving function highway pilot, 8th International Munich Chassis Symposium. Munich, Germany.

LMS Imagine.Lab Amesim: https://www.plm.automation.siemens.com/en/products/lms/imagine-lab/amesim/

PreScan: https://www.tassinternational.com/prescan