A complete modular framework for developing and testing automated driving controllers

Ray Lattarulo * **Joshué Pérez** * **Martin Dendaluce** *

* *Tecnalia Research and Innovation, Derio, Vizcaya, Spain, 48160. (e-mail: {rayalejandro.lattarulo, joshue.perez, martin.dendaluce}@tecnalia.com).*

Abstract: Intelligent vehicles have been improved their highly and fully automated capacities in the last years. Most of the developments are due to the fast evolution of embedded systems for the acquisition, perception and communication on-board modules. However, the fast growing of automated vehicles market demands modern tools for validation, integration and testing of these new embedded functionalities, specially related to Advanced Driving Assistance System (ADAS). In this paper, a testing methodology for validation of planning and control algorithms for current and future automated vehicles is presented. The modularity and adaptability have been considered in the design of the proposed method. It is based on a software tool for vehicle modeling, called Dynacar, which allows a good trajectory definition, cooperative maneuvers interaction and virtual validation. Different kinds of vehicles, scenarios (i.e.: urban, interurban, highways with different environment conditions) and controllers can be tested. Moreover, Hardware-In-the-Loop configuration (i.e. electric motors) can be also tested. Simulations results show a good performance in the implementation and configuration of urban scenarios, using different controllers in the proposed framework.

Keywords: Intelligent transportation systems, Control architectures in automotive control, Simulation, Control validation, Control Testing.

1. INTRODUCTION

Nowadays, Advance Driver assistance Systems (ADAS) and automated driving capacities are becoming a reality in mass-produced vehicles Policy (September 2016). These systems help on the reduction of accident rates on highway and urban scenarios, caused due to human driver errors. The validation of these new functionalities needs modern tools to integrate and test new scenarios and vehicles.

Different simulations environment are more related to the acquisition, perception and communications stages, as in Bounini et al. (2015). Other approaches are focused in model identification (Dias et al. (2015)), based on real data from an instrumented vehicle. However, from the control point of view, only linear longitudinal model have been considered for low speeds controllers, where the main applications include parking and urban adaptive cruise control.

In the automotive industry, the virtual testing helps considerably in the cost reduction for the validation and testing stages of the new intelligent functionalities, as have been presented in the DESERVE project (Kutila et al. (2014)). The goal is to develop and adjust control algorithms without the need of testing them in real scenarios all the time, due to legislation limitations.

Real data set testing will always improve the validation of any algorithm to be tested in the vehicle. For this reason, some partial validation can be performed with Hardware and Model-In-the-Loop solutions. It will help to reduce the gap between real and virtual behavior.

Since the development process and the validation of the embedded controller are becoming more complex, a solution for testing the decision makers, path planning and control modules is needed. Based on the general architecture for automated vehicles used by most of the researcher groups and the automotive industry around the world González et al. (2015), in this paper, a testing framework for the validation of decision, planning and control algorithms is presented. Our approach considers the 6 main stages: acquisition, perception, communication, decision, control and actuators, and the automation levels defined in SAE J3016 standard SAEJ3016 (2014) for automated driving. The solution allows a real time performance, based on a software and hardware module independence. This advantage is very useful for testing some ADAS functions with on-board ECUs for modern vehicles, and even to emulate the setup from data acquisition sensors, as in Differential GPS. Our approach is useful to validate, first in simulation and then using HIL methods, the most common individual and cooperative maneuvers (i.e.: overtaking, intersections, merging, roundabouts, platooning, etc).

The rest of the paper is organized as follow: a review simulation tools for automated vehicles is explained in Section II. Details of the implementation of the Dynacar simulator are also given. Section III explains the proposed testing framework. Validation and tests results are described in

Section IV. Finally, conclusions future works are listed in Section V.

2. CURRENT STATE OF MODELING AND SIMULATION ENVIRONMENTS FOR ADAS AND AUTOMATED VEHICLES

Vehicle simulators are very valuable for validation and testing of new functionalities for automated driving. In order to increase security and safety in new algorithms, more accurate dynamic models of vehicle in simulation environments are needed in the automotive research and industry.

In order to develop better decision and control features for automated vehicles, a precise multi-body model to describe the vehicle dynamics in different scenarios is still missing. Robust dynamic models, based on a correct environment modeling, will lead a reduction of development cost of the new ADAS. In this section, a description of the current simulator available on market is presented.

2.1 Civitec Pro-Sivic

This tool has been designed as a sensor simulator by the company CIVITEC, mainly oriented to perception and ADAS developments. It gives the opportunity to simulate a variety of complex scenarios like intersections, roundabouts, multiple vehicles and pedestrians on the road, and changes in the weather conditions rain, fog, snow, brightness and others. The parameters are adapted taking into account the behaviors of sensors and weather. Moreover, it has the potential of being connected with RTMaps platform to test new algorithms, for example in data fusion and control. There are some technologies that have been tested on this platform on last years. Real-time road lane detection and tracking is one interesting contribution Bounini et al. (2015). Another application is the use of the platform Simulink/Matlab to communicate it with the Pro-SIVIC simulator to test cooperative algorithms, like CACC (Cooperative Adaptive Cruise Control), on automated vehicles Bounini et al. (2014).

2.2 CarSim and TruckSim

CarSim and TruckSim are products of the Mechanical Simulation Company. They were developed to use dynamics vehicle models, for passenger cars and light trucks, respectively. Those offer some capabilities like software-in-the-loop (SiL), model-in-the-loop (MiL), hardware-in-the-loop (HiL), driver-in-the-loop (DiL), supporting of vehicle sensors and vehicle-to-vehicle (V2V) communication. Additionally, they have a standard interface to Matlab/Simulink. This simulator is main oriented to validation and ADAS, i.e.: obstacle avoidance algorithm using model predictive control Abbas et al. (2014), robust control for in-wheel motor vehicles Gaspar et al. (2014) and dynamic trajectory generation Hafner and Pilutti (2014).

2.3 IPG Automotive CarMaker and TruckMaker

PreScan of Tass international is a simulator that is specialized on sensor modeling like GPS, vision, laser, radar, accelerometer and odometry. It has the capability of manage n-vehicles and n-sensors in each scenario. It is also based on Matlab/Simulink.

Some of the most important works made on this platform are a full spectrum camera simulation for reliable virtual development and validation of automated driving applications Molenaar et al. (2015), among others.

2.4 Tass international PreScan

PreScan of Tass international is a simulator that is specialized on sensor modeling like GPS, vision, laser, radar, accelerometer and odometry. It has the capability of manage n-vehicles and n-sensors in each one. It is also based on Matlab/Simulink.

Some of the most important works made on this platform are a full spectrum camera simulation for reliable virtual development and validation of automated driving applications Molenaar et al. (2015), among others.

2.5 SCANeR

SCANeR Studio is a software suite for Human-In-the-Loop driving simulations; it is developed by OKTAL, and is based on works of the Vehicles Simulation and Perception research group of Renault and works of SERACD.

Several European projects have been used as background for the development of the software, examples are Prometheus, TRaCS (TRuck and Coach Simulator), CARDS (Comprehensive Automobile R&D Simulator).

2.6 SPEOS VRX OPTIS

Based on the previous experience in SPEOS driving simulator platform, OPTIS has developed the SPEOS VRX environment for ADAS and Automated Driving testing. This tool is focused on the acquisition of different sensor embedded in the vehicles. It includes camera, LIDAR, infrared sensors, among other. They also provide a 3D view of the car environment in 360 degrees.

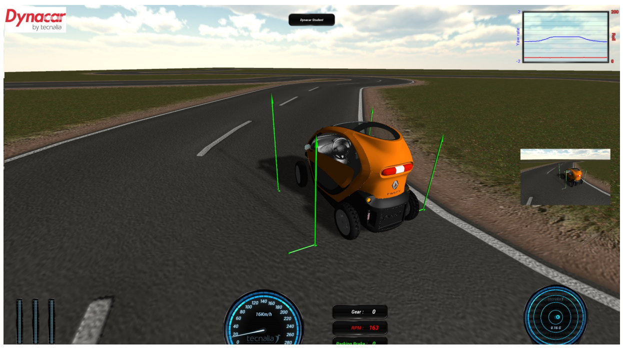

Fig. 1. Dynacar by Tecnalia

2.7 Tecnalia Dynacar

Dynacar (figure 1) is a simulation tool developed by Tecnalia which provides a real-time vehicle model covering multiple domains. It focuses on vehicle dynamics, providing a high-fidelity vehicle physics simulation basing a

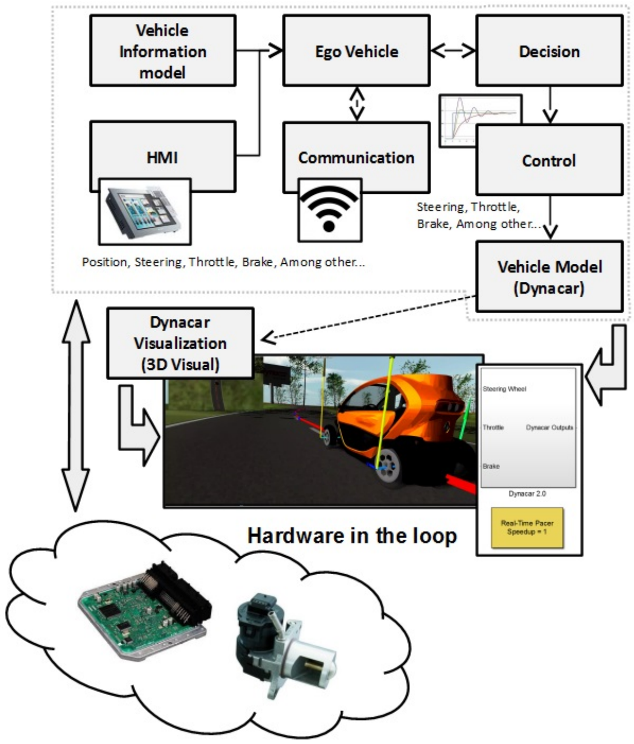

Fig. 2. Architecture framework for automated Driving

multibody vehicle dynamics model. This is combined with a Pacejka tire model, and submodels for elements like the engine, transmission, steering system, braking system, aerodynamics, etc Iglesias (2015). Moreover, it is used in the Electric-Electronic architecture of the vehicle. Here subsystem models (from ECUs to components) can be connected for development of control functions.

Dynacar permits real-time and accelerated time simulations. The real-time capability is very valuable, as, combined with its notable modularity and interfacing options, it permits to execute tests with driver-in-the-loop (DiL) and hardware-in-the-loop (HiL) setups, for instance for ECU (Electronic Control Unit) development or also motor test bench testing Aguinaga et al. (2013), integrated into a Simulink blocks.

This work is based on Dynacars capabilities, using them to implement automated driving maneuvers. A testing methodology for the validation of control algorithms for future automated vehicles is presented. The modularity and adaptability has been exploited in the design of the proposed method based on the a general control architecture for automated driving presented in González et al. (2015). It enables a good trajectory definition, cooperative maneuvers and virtual validation with different kind of vehicles and scenarios.

3. PROPOSED TESTING ARCHITECTURE

The proposed framework is shown in figure 2. It includes a precise dynamic model of the vehicle to test automated capacities, as: real time path generation, control, communication and other algorithms that could be embedded in a ECU.

Our approach have been implemented on Simulink Matlab, which is used on the automotive research due to the process data and also the possibility to include C-based libraries on the implementation of new blocks (figure 3).

The main components of the automated driving environment are:

3.1 Vehicle Information Model

This module is responsible to adapt the signals of position, speed, among others from the simulator. It integrates the same magnitude and units, to be received on the real platform from the different sensors (Differential GPS, Odometry, lasers, among others).

One of the most relevant features of the proposed framework is related to scalability of data from the simulator to the real platform.

3.2 HMI

The human machine interface (HMI) is responsible to communicate real-time information and configuration parameters to the modules as in the real vehicle. It represents the bridge of interaction between the automated driving platform with the user or driver.

3.3 Communication

This module incorporate the possibility of simulate a V2X (vehicle to X) communication that is relevant in the process of testing cooperative maneuvers and more complex scenarios.

The platform is useful to test the actions that should be performed for the vehicle on presence of data lost or reduction in signal quality from other vehicles or infrastructure.

3.4 Ego-vehicle

Ego-vehicle module is responsible to gather data from the vehicle information model, HMI and communication modules to the decision and control modules. This module will pack the information and configuration parameters of the vehicle for the rest of the following modules.

3.5 Decision

The decision module is separated in two blocks: Global and Local planners, based on the review of the path planner techniques for automated vehicles González et al. (2015). The first one is able to give the route to be followed, and the second to generate a real time dynamic planning.

Global planner The global planner is in charge of reproduce the route that will be followed by the vehicle. It considers the information of the starting and the ending points, and even the possible changes of routes through the HMI.

This path should be precise to generate a soft local planner. Special configurations of urban environments should be considered as intersection, roundabouts and merging.

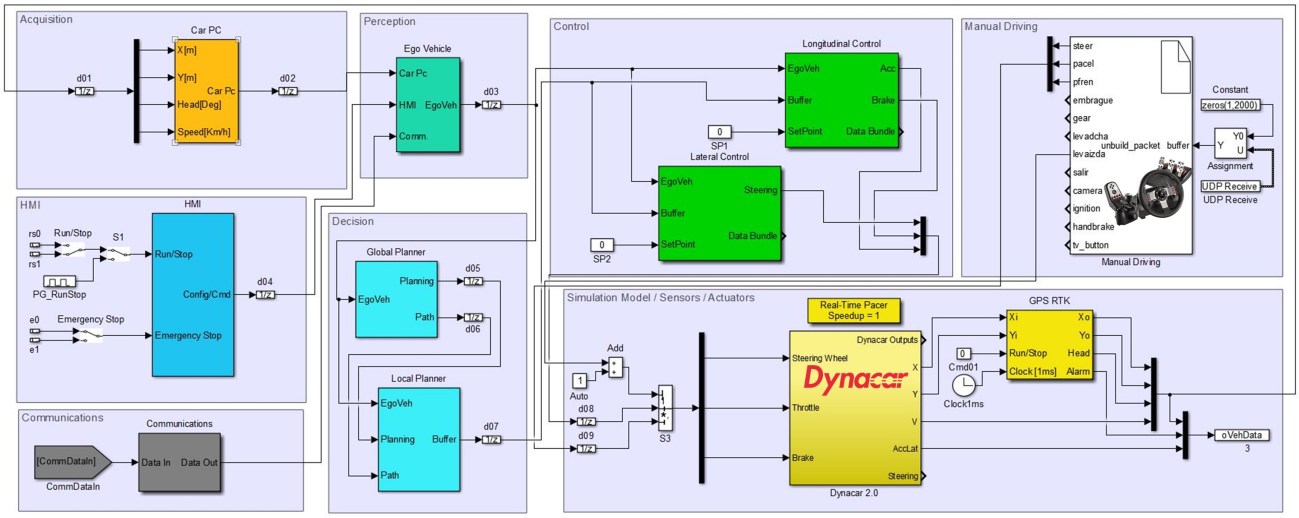

Fig. 3. Automated driving testing tool and interaction with 3D model

Local planner Based on the information coming from the global planner, a soft trajectory is generated using parametric curves as in Pérez et al. (2013a). Here, the aim is to reduce the curvature change between the straight segments and curves.

The local planner considers the possibility of generate continuous trajectories in case of an unexpected scenario as an obstacle avoiding, merging, among others. Intelligent dynamic algorithm for trajectory generation has been done as in Rastelli et al. (2014).

Bezier curves are often used for real time parametric generation, as in González et al. (2015) following the equation 1.

$$B_t = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} * t^n * P_i \quad (1)$$

These curves are easy to implement, and they reproduce a soft curvature with a reduced computational level.

Planning buffer A communication buffer between the decision and control module is defined to generate sudden change in the horizon view of the local planner. It will optimize and increase the response time of control law calculation. Any unrealistic trajectory (i.e.: if an obstacle is detected in front) will be delate before to be sent to the control mode.

3.6 Control

This module receive the buffer information from the decision stage, and it is responsible of the lateral (steering) and longitudinal (acceleration and breaking) control.

For instance, PID controllers for lateral and longitudinal response can be used. In the case of the lateral control, the lateral, the angular error and the curvature of the path as González et al. (2014). Other techniques, as fuzzy logic as in Pérez et al. (2013a), model predictive control, among others, will be easily integrated in our framework.

3.7 Vehicle model and visualization

The Dynacar multibody simulator has been used to model the real vehicle. Additionally, the visual system provided by this tool lets monitoring in real time the algorithms implemented.

3.8 Hardware In the Loop configuration

The existing Dynacar platform was conceived to enable diverse HiL setups, including automated driving algorithm developments, through a physical ECU. It can be used to perform validation tests, which will be later deployed on the real vehicle by connecting the ECU to a virtual Simulink-based environment and subcomponent models, through the Dynacar framework. As this includes a multibody model, the reactions and feedback signals are very realistic. Furthermore, the ECU can interact with Simulink over a physical CAN bus, using the real vehicles DBC (CAN database file), therefore handling the messages as they will occur on the car.

4. RESULTS AND VALIDATION

4.1 Tests of planning algorithm

Parametric curves have been implemented on the local planning generation to create a continuous path for simulation. Figure 4 shows an urban scenario, including intersection and roundabouts. The black markers show the global map, modeling five roundabouts and ten intersections. On blue color are plotted the local planning points, and the red line represent the tracking of the vehicle.

4.2 Tests of lateral control algorithm

To validate the model, some tests have been made on the lateral control. Five different controllers were used, based on lateral error, angular error and the curvature as parameters.

Figure 5 and 6 shows the controllers tested on different speeds (15 and 25 kilometers per hour). The upper part of

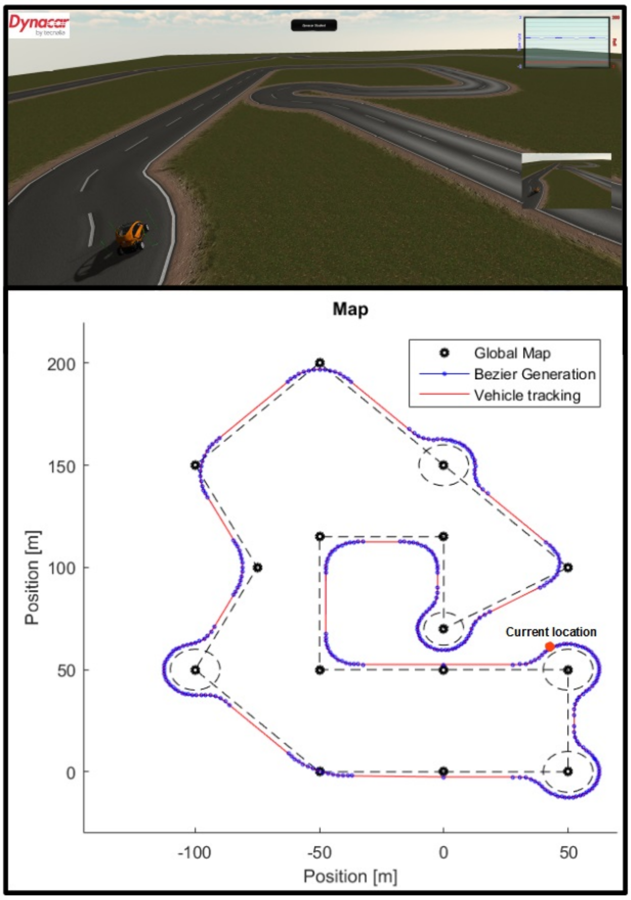

Fig. 4. Urban scenario route.

both figures shows the evolutions of the lateral error, the higher speed reference value, the higher lateral error. The other graphics show the angular error and steering output in each experiment.

Proportional (C1) The proportional controller shows the worst error obtained. Based on figures 5 and 6 this controller is represented with a black line.

Proportional-Integrate (C2) The PI controller is represented by the red color line. It has a better result, reducing the lateral error in comparison with the proportional controller but increase the presence of overshoots and the time to stabilize is longer.

Proportional-Integrate-Derivative (C3) The PID controller improves the results of the first two controllers, with the reduction of the lateral error and the overshoots produced by the integrate part. This is represented with a green color line.

Lateral error and angular error equation (C4) This controller is based on Pérez et al. (2013b), based on the lateral and angular errors, with a proportional and derivative action, respectively. K_1 and K_2 are the gains fixed manually on the vehicle. In this sense, the equation used in the fourth control law is:

$$C_v = K_1 * e_{lat} + K_2 * e_{ang} \tag{2}$$

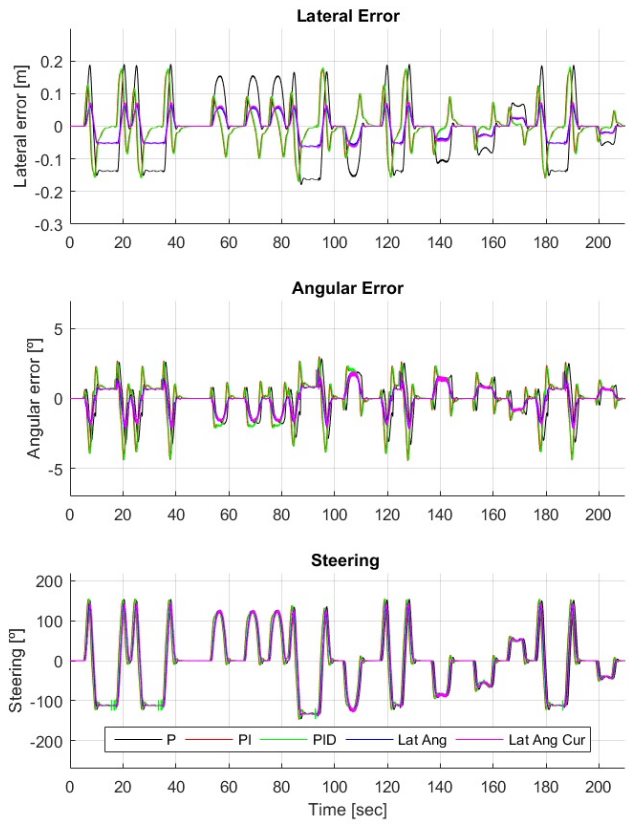

Fig. 5. Data recorded at 15 km/h.

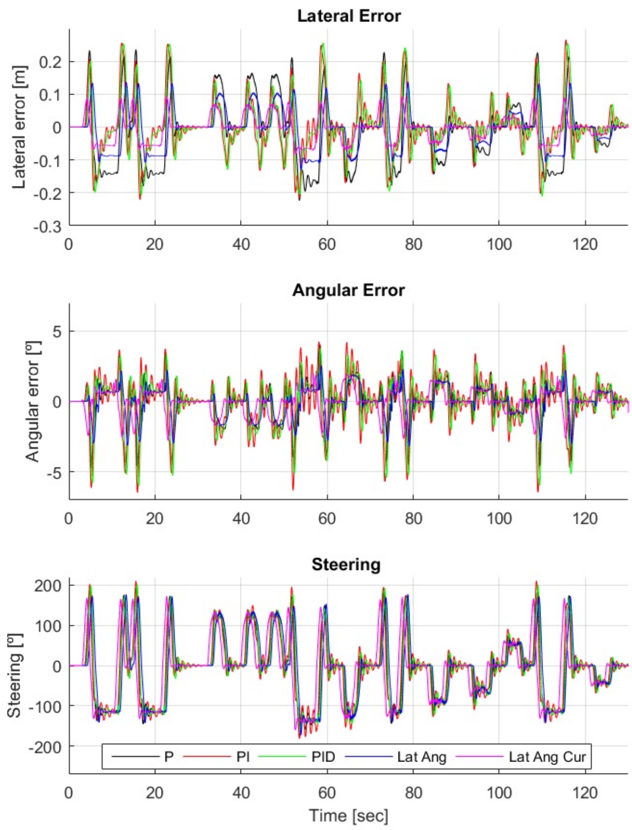

Fig. 6. Data recorded at 25 km/h.

This presents results similar to the PID controller, but reducing even more the lateral error. The tuning process was even easier than in the PID case, because only two variables were used. It was plotted with a blue color line.

Lateral error, angular error and curvature equation (C5) This controller were used to tune real vehicles in Rastelli et al. (2014). An improvement of the method before, it is showed as follow:

$$C_v = K_1 * e_{lat} + K_2 * e_{ang} + K_3 * Curvature \quad (3)$$

This equation includes a third parameter. The curvature improves even more the lateral error in comparison with the PID. This last controller is represented with a pink color line on figure 5 and 6.

Table 1. Average lateral error for different controllers.

	Lat. error 15km/h $[m]$			**Lat. error 25km/h** $[m]$		
	Ave.	**Med.**	**Max.**	**Ave.**	**Med.**	**Max.**
P	0,063	0,049	0,19	0,067	0,044	0,24
PI	0,036	0,016	0,17	0,052	0,027	0,27
PID	0,036	0,015	0,18	0,051	0,024	0,26
C4:	0,023	0,018	0,07	0,040	0,028	0,14
C5:	0,024	0,019	0,07	0,026	0,018	0,09

Human intervention in the control loop Sharing control and arbitration are increasing the interest on our research field. In this sense, the current architecture gives the opportunity of switching between both automatic and manual control to test this new approaches.

4.3 Longitudinal control

The test presented in figures 5 and 6 were performed on constant speed, and based on a PID controller, for both speeds.

Our approach is capable to deal with profile speeds adapted to the curvature of the path. Figure 7 shows the behavior with consecutive curves and roundabouts, using a *fairy uncomfortable* lateral acceleration (based on Labakhua et al. (2008)).

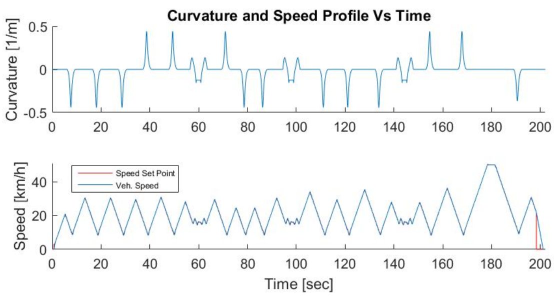

Fig. 7. Speed profile for urban scenarios, based on the curvature

5. DISCUSSIONS AND CONCLUSIONS

Figure 4 shows how an urban scenario was modeled for testing a planning algorithm considering urban scenarios (with multiple roundabouts and intersections) and, the dynamical characteristics and conditions of a Renault twizy (Vehicle model used). However, this platform is full capable to change to another different vehicle dynamics.

Multiple control algorithms were tested, mainly involving lateral, angular and curvature errors. On table 1, a comparison of the lateral error (average, median and maximum value) for the controllers explained on sections before is shown.

These results demonstrate how lateral control algorithms are tested on the virtual platform using low (15 km/h) and average (25 km/h) speeds in urban areas, and how each controller is compared among of them. Moreover, a speed profile generator based on the path curvature was presented. The current framework is capable to test high speed scenarios. In the paper, we present different control techniques to validate the automated capabilities of an electric vehicle at different speeds in urban scenarios.

6. FUTURE WORKS

On the last two years, the developments on automated driving functions have been increased significantly. On the current approach, a complete framework for automated driving testing has been presented with different controllers and path planning algorithm in urban scenarios.

The current approach allows testing different blocks for the validations of automated driving vehicles. Safety and security will be considered in future algorithm implementation to have a higher level of integrity, as addressed by the ISO-26262 standard.

Lateral acceleration and comfort will be considered in the current platform to validate them for embedded longitudinal and lateral controllers. Fault injection for safety and controllability evaluation in different scenarios can be also tested. Finally, communication among vehicles and infrastructure will be tested considering dangerous situations like package data lost or corrupted information.

This work is developed in the context of EU and national projects. Next steps are related to the implementation and validation of different scenarios for real automated vehicles (i.e.: two lines intersection, overtaking, cooperative roundabouts, etc).

ACKNOWLEDGEMENTS

Authors wants to thank to the H2020 UnCoVerCPS Project and ESCEL Project ENABLE-S3 for its support in the development of this work.

REFERENCES

Abbas, M.A., Milman, R., and Eklund, J.M. (2014). Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. *IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1 – 6.

Aguinaga, I.I., Sandi, A.M., and Rodriguez, A.P. (2013). Vehicle modelling for real time systems application. the virtual rolling chassis. *Revista Dyna*, 88, 206 – 215.

Bounini, F., Gingras, D., Lapointe, V., and Gruyer, D. (2014). Poster: Realtime simulator of collaborative autonomous vehicles. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1991 – 1994.

Bounini, F., Gingras, D., Lapointe, V., and Pollart, H. (2015). Autonomous vehicle and real time road lanes detection and tracking. *IEEE Vehicle Power and Propulsion Conference (VPPC)*, 1 – 6.

Dias, J.E.A., Pereira, G.A.S., and Palhares, R.M. (2015). Longitudinal model identification and velocity control of an autonomous car. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 776–786. doi: 10.1109/TITS.2014.2341491.

Gaspar, P., Bokor, J., Mihály, A., Szabó, Z., Fulep, T., and Szauter, F. (2014). Robust reconfigurable control for in wheel motor vehicles. *IEEE International Electric Vehicle Conference (IEVC)*, 1 – 6.

González, D., Pérez, J., Lattarulo, R., and Milanés, V. (2014). Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1430 – 1435.

González, D., Pérez, J., Milanés, V., and Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17, 1135 – 1145.

Hafner, M. and Pilutti, T. (2014). Dynamic trajectory planning for trailer backup. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2501 – 2506.

Iglesias, I. (2015). Herramienta para acelerar el desarrollo de nuevos sistemas y controles para automocin. *Revista Automtica e Intrumentacin*, 478, 45 – 49.

Kutila, M., Pyykonen, P., van Koningsbruggen, P., Pallaro, N., and Pérez-Rastelli, J. (2014). The deserve project: Towards future adas functions. In *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014 International Conference on*, 308–313. doi:10.1109/SAMOS.2014.6893226.

Labakhua, L., Nunes, U., Rodrigues, R., and Leite, F.S. (2008). Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation. *Informatics in Control Automation and Robotics*, 15, 169 – 182.

Molenaar, R., van Bilsen, A., van der Made, R., and de Vries, R. (2015). A full spectrum camera simulation for reliable virtual development and validation of adas and automated driving applications. *IEEE Intelligent Vehicles Symposium (IV)*, 47 – 52.

Pérez, J., Godoy, J., Villagrá, J., and Onieva, E. (2013a). Trajectory generator for autonomous vehicles in urban environments. *IEEE International Conference on Robotics and Automation (ICRA)*, 409 – 414.

Pérez, J., Nashashibi, F., Lefaudeux, B., Resende, P., and Pollard, E. (2013b). Autonomous docking based on infrared system for electric vehicle charging in urban areas. *Sensors journal*, 2645 – 2663.

Policy, F.A.V. (September 2016). Accelerating the next revolution in roadway safety. Technical report, NHTSA, U.S. Department of Transportation.

Rastelli, J.P., Lattarulo, R., and Nashashibi, F. (2014). Dynamic trajectory generation using continuous curvature algorithms for door to door assistance vehicles. *IEEE Intelligent Vehicles Symposium (IV)*, 510 – 515.

SAEJ3016 (2014). Taxonomy and definitions for terms related to on road motor vehicle automated driving systems. Technical report, SAE International.