

Research Article

Low Speed Longitudinal Control Algorithms for Automated Vehicles in Simulation and Real Platforms

Mauricio Marcano , José A. Matute, Ray Lattarulo, Enrique Martí, and Joshué Pérez

Tecnalia Research & Innovation, Derio, Spain

Correspondence should be addressed to Mauricio Marcano; mauricio.marcano@tecnalia.com

Received 24 November 2017; Accepted 19 February 2018; Published 29 March 2018

Academic Editor: Victor M. Becerra

Copyright © 2018 Mauricio Marcano et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advanced Driver Assistance Systems (ADAS) acting over throttle and brake are already available in level 2 automated vehicles. In order to increase the level of automation new systems need to be tested in an extensive set of complex scenarios, ensuring safety under all circumstances. Validation of these systems using real vehicles presents important drawbacks: the time needed to drive millions of kilometers, the risk associated with some situations, and the high cost involved. Simulation platforms emerge as a feasible solution. Therefore, robust and reliable virtual environments to test automated driving maneuvers and control techniques are needed. In that sense, this paper presents a use case where three longitudinal low speed control techniques are designed, tuned, and validated using an in-house simulation framework and later applied in a real vehicle. Control algorithms include a classical PID, an adaptive network fuzzy inference system (ANFIS), and a Model Predictive Control (MPC). The simulated dynamics are calculated using a multibody vehicle model. In addition, longitudinal actuators of a Renault Twizy are characterized through empirical tests. A comparative analysis of results between simulated and real platform shows the effectiveness of the proposed framework for designing and validating longitudinal controllers for real automated vehicles.

1. Introduction

Nowadays, traffic jam is a well-known cause of the increase in travel times, accidents, and fuel consumption in urban areas. In Europe, more than 200,000 traffic jams have been identified across 123 cities in September 2016 [1]. This study shows that the annual cost associated is more than €4.9 billion. This problem is being currently tackled by authorities, industry, and research centers.

Automated vehicles and Advanced Driver Assistance Systems (ADAS), as part of the Intelligent Transportation Systems (ITS), are considered as short and medium term solutions for the previous problem. Based on the last ERTRAC (European Road Transport Research Advisory Council) report [2], traffic jam assist systems are partially available in the market. However, most of these solutions are restricted to high speed (i.e., more than 30 Km/h), with automation level 2, based on the SAE J3016 standard [3].

The most important advances in this area include Cruise Control (CC) in high speed [4], dynamic stability control

[5], pedestrian detection systems combined with collision avoidance [6], and semiautonomous parking [7].

In spite of the aforementioned advances, validation of some functionalities is a complex and tough task. Based on the classical testing proposed in [8], automated vehicles need to drive around 275 million Km to demonstrate a performance superior to humans. This would take around 12.5 years considering a fleet of 100 vehicles driving nonstop. In contrast, the self-driving car project [9] from Waymo has only reached 5.6 million miles driven in real conditions [10]. However, they have recently announced a simulation environment able to drive 12.8 million Km per day, where they have drove 4 billion Km in total. The previous is aligned with the suggestions made in [8], encouraging simulation testing to validate automated vehicles functionalities.

Currently, most simulation tools are focused on computational perception [11], test benches for hardware in the loop testing [12], and evaluation of human acceptance regarding different ADAS [13]. However, from the control point of view, few solutions have been implemented.

In [14] authors present a comparative study for low speed longitudinal controllers, based on a gasoline-propelled vehicle, using classical and intelligent control techniques; nonetheless, the controllers were designed and validated directly in the real vehicle and no comparison against simulations was presented. Other authors worked in throttle and brake controllers in vehicle following applications presenting real and simulated results [15] based on fixed and variable gains PID controllers, but without an extensive comparison between the platforms.

This paper presents a use case for Cruise Control where three longitudinal low speed control techniques are designed, tuned, and validated using an in-house simulation framework and later applied in a real vehicle. Control algorithms include a classical PID, an adaptive network fuzzy inference system (ANFIS), and a Model Predictive Control (MPC). The experimental procedure aims to validate the simulation platform as an effective tool for ADAS functionalities development, able to save time and cut costs by reducing the development effort that involves working with the real vehicle.

The rest of the paper is organized as follows: Section 2 explains the vehicle model and its parameters; additionally, an empirical characterization of longitudinal actuators in real test platform is described. Section 3 describes the algorithms applied on low speed control. Tests and results of the three controllers implemented in both simulation and real vehicle are presented in Section 4. Lastly, conclusions and recommendations for future works are given.

2. Description of Platforms

Longitudinal speed controllers are implemented in both simulation and real vehicle. This section offers a description of both platforms based on an automated driving general framework. First, a brief summary of the global architecture is presented, followed by an explanation of the instrumentation and automation of Renault Twizy. Lastly, the simulation environment used to emulate the real vehicle through a mathematical model will be detailed.

2.1. Automated Driving General Framework. The control algorithms were developed under the automated driving framework presented by David González and Joshué Pérez in [16]. This architecture has been presented in previous works, including a validation of lateral controllers in simulation [17]. The global frame is composed of six blocks that abstract a generic processing pipeline for automated driving applications, as detailed below:

- (1) *Acquisition*: this module gathers information coming from on-board sensors and does preprocessing tasks such as time stamping, outlier rejection, or reference system transformations. Current implementation accepts data from a variety of sources, including differential GPS, Inertial Measurement Unit (IMU), vision sensors, and others. Low level CAN communication with vehicle internal system is also considered within this module.
- (2) *Perception*: perception module is in charge of mixing raw data received from acquisition block to generate

information useful for automated driving tasks. Generated data includes ego-vehicle estimation, obstacle detection/classification, lane detection, and traffic signal recognition. This module features machine learning, deep learning, and sensor fusion algorithms that provide an accurate, robust, and consistent view of the world. Those techniques allow reducing the uncertainty of original measure and characterize its associated error.

- (3) *Communication*: communication module provides information coming from external entities (vehicles and infrastructure). From a wider perspective, this module establishes a bidirectional information flow to manage the integration of the vehicle in a cooperative driving environment.
- (4) *Decision*: decision module determines the trajectory and speed plan to be followed by the vehicle. This is done in three consecutive steps: the global planner generates a feasible trajectory from the available road map. Local planner refines it to optimize different parameters (e.g., comfort, safety), through minimization of curvature and its derivatives, acceleration, and jerk. This is achieved using different types of curves, such as Joshué Pérez et al. [18]. The last step is behavioral planner, which adapts the route to face dynamic events as obstacles in the road, other vehicles' maneuvers, or unexpected traffic restrictions.
- (5) *Control*: it generates normalized control actions over longitudinal and lateral vehicle actuators, to ensure that the plan produced by decision module is effectively accomplished. This work contributes to this specific block, along with the actuation module presented below.
- (6) *Actuators*: abstraction layer transforms normalized control outputs to real action signals over the actuators: throttle, brake, and steering wheel. It enables transparent switch between real and simulated environments.

As shown in Figure 1, the general framework is independent of the test platform, making this approach a versatile tool for testing control algorithms for automated driving applications in both real and simulated vehicles. This work is based in Matlab/Simulink.

2.2. Real Test Platform. The experimental platform is a Renault Twizy 80 (see Figure 1), a 4-wheel electric vehicle capable of reaching a maximum speed of 80 [Km/h] with vehicle and motor parameters depicted in Table 1. It has been instrumented to control the steering and the pedals through CAN bus network.

As shown in Figure 2, the three systems are managed by different electronic devices and actuators. First, the steering wheel is rotated by a stepper motor controlled through PWM signals. Throttle command is communicated to vehicle's Engine Control Unit (ECU) through an analog voltage signal between 0 and 10 VDC. The brake pedal is coupled with a mechanical linear actuator. All the instrumentation is

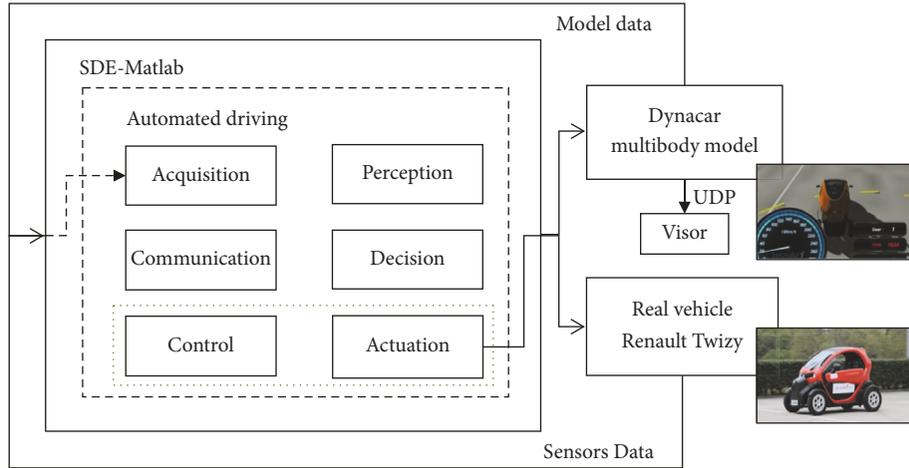


FIGURE 1: Test platforms and general framework.

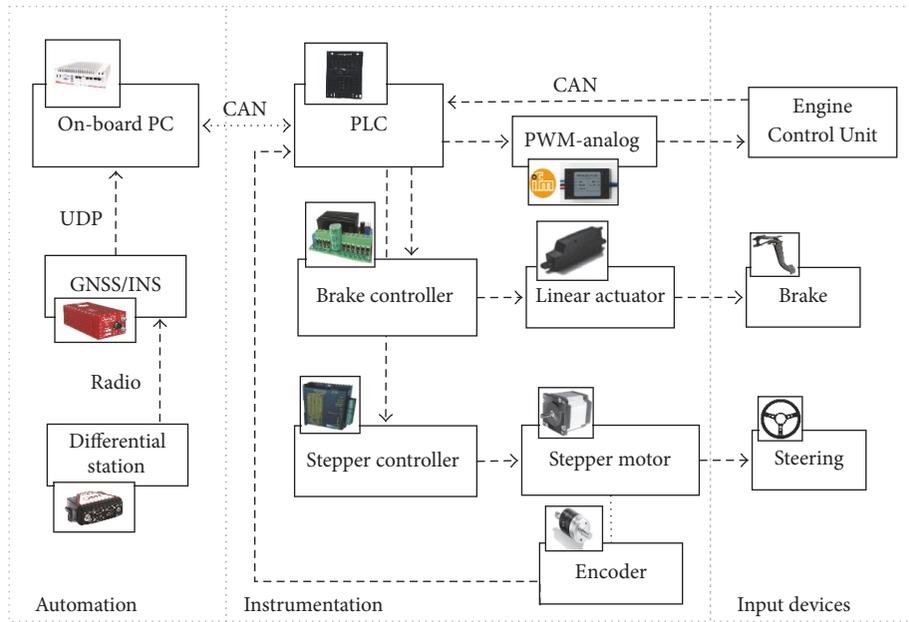


FIGURE 2: Instrumentation and automation of a Renault Twizy.

controlled by a Programmable Logic Controller (PLC) that acts as an interface between the actuators and the central computer. The former executes the automated driving general framework and the control algorithms. It is also in charge of collecting the data from the external sensors such as GPS and IMU. More details on sensors and devices are listed in Table 2.

2.3. Simulation Test Platform. The virtual platform is based on Dyncar, an integrated solution for design of electric and hybrid vehicles that features a vehicle physical model based on multibody formulation, making use of relative coordinates and semirecursive equations of motion based on velocity transformation. Suspensions are considered as macro-joints, and the behavior is modeled using lookup tables. As shown in Figure 3, local Cartesian coordinates of the chassis frame are located at the middle of the frontal track

width (C), cardan angles providing wheels orientation with respect to the chassis frame are located at the knuckles (K), and kinematic expressions for the macro-joints consider the position, velocity, and acceleration levels of the wheels (W) [19, 20].

The powertrain system includes different engine and transmission models. The electric motor is modeled defining values of motor torque [N-m], motor speed [rpm], and throttle pedal position [in%] through a lookup table and interpolating linearly during simulations. It is possible to introduce regeneration providing a constant braking torque as long as the throttle pedal is fully released and the motor speed is over certain value. A clutch is modeled using a linear interpolation within a predefined table between clutch's displacement and torque transmitted; however a fixed clutch was selected due the real platform characteristics. A

TABLE 1: Real platform parameters.

| | |
|--|---|
| Mass (kg) | 611.500 |
| Dimensions (l) | 611.500 |
| CG location x, y, z (m) | -0.928, 0.000, 0.488 |
| Wheelbase (m) | 1.686 |
| Track width (m) | 1.094 |
| Inertia I_x , inertia I_y , inertia I_z (kg-m ²) | 243.175, 430.166, 430.166 |
| Front wheel radius (m) | 0.265 |
| Rear wheel radius (m) | 0.281 |
| Motor type | Three-phase asynchronous w/regeneration |
| Power | 11 [hp] between 220 and 785 [rad/sec] |
| Maximum torque | 57 [N-m] between 0 and 220 [rad/sec] |
| Transmission reduction | 1:9.23 |

TABLE 2: Sensors and electronic devices.

| | |
|----------------------------|---|
| Position and inertial data | GNSS-aided IMS + base station |
| On-board computer | Fanless, i7-6700TE, 32 GB RAM |
| Brake system | Linear actuator, 750 N 20 A DC motors driver |
| Steering system | High torque hybrid stepping motor Magnetic encoder with CANopen 20-80 V stepping motor driver |

gearbox is modeled calculating the output torque multiplying the input torque by transmission ratio and gear efficiency coefficient. The final traction moment is calculated after considering a fixed differential sending the propulsion flow to the rear wheels [21].

The brake system consists in a simplified model (see (1)) which combines the brake pedal travel (E_b) [°], maximum pressure on the circuit (P_{max}) [MPa], percentage of braking pressure ($K_{i,j}$) [°], and brake pad capacity ($C_{i,j}$) [N-m/MPa] to obtain the braking torque on the wheels ($M_{i,j}$) [N-m]. Subindex i refers to the location of each wheel (i = front/rear, j = left/right) [21].

$$M_{i,j} = E_b P_{max} K_{i,j} C_{i,j}. \quad (1)$$

2.4. Characterization of Longitudinal Actuators. Throttle and brake pedal positions specify the capacity of acceleration and deceleration of the vehicle. Both simulation and real test platform have a signal control range that goes from 0 to 100% of pedals location. In the real platform, the intermediate instrumentation devices (acceleration potentiometer signal and brake linear actuator) introduce undesired effects such as biases, scaling, or delays that play an important role in low level of control framework, as well as the way longitudinal actuators act over the pedals (Figure 2).

In order to approximate the actuator behavior in simulations, moving tests on real test platform were evaluated at different throttle and brake pedals positions. Each test starts from a stopped position with a 9 s period of constant acceleration, followed by 4 s with no action over pedals and

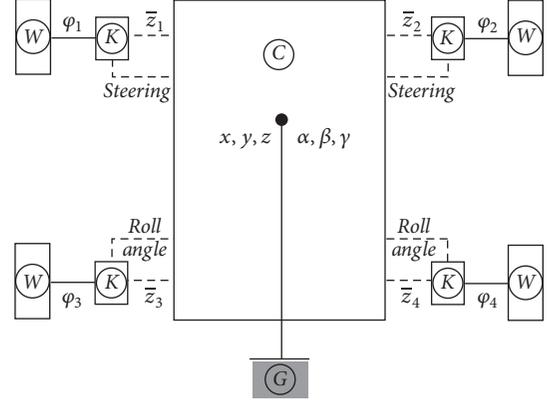


FIGURE 3: Dynacar multibody formulation [20].

finishing with 2 s of constant braking. A total of 7 different tests were performed, varying both pedals position from 40% to 100% of their maximum displacement in 10% steps (Figure 4(a)). These three periods were useful for evaluating actuator's influence on traction, regeneration, and braking.

The inertia of physical devices as actuators or the brake hydraulic system strongly affects the longitudinal behavior of the vehicle. Additionally, an electrical lumped lag can be found after pushing the throttle pedal due a combination of devices within the control framework and the vehicle. Therefore, mechanical and electrical lags (τ) during traction and braking observed in moving tests (Figure 4(b)) can be modeled in simulations as a discrete transfer function considering a sampling time (t_s):

$$H(z) = \frac{(t_s)}{(\tau z + (t_s - \tau))}. \quad (2)$$

Average acceleration values during traction and braking periods were normalized in $[-1; 1]$, obtaining curves that represent the real influence of actuators over pedals for different signal control values sent from on-board computer (Figure 5). Throttle response has a linear behavior due a direct connection with the potentiometer. In contrast, braking response has a polynomial behavior determined mostly by the orientation of the linear actuator with respect to brake pedal. Additionally, both pedals are also affected by range and scales of voltages previously programmed in the PLC. These results are very useful for resembling truthfully the longitudinal behavior of the vehicle during simulations, making an equivalence between the signal control and the actuators influence over the vehicle.

3. Design of Control Algorithms

This section presents the design of three longitudinal controllers for low speed CC; these are based on fixed gains (PID), trained data (ANFIS), and a predictive model (MPC). The algorithms depend on the speed error calculated as the difference between controller reference and the actual speed. In simulation inertial data is provided by the virtual model, while the experimental test uses the GNS/INSS sensor information.

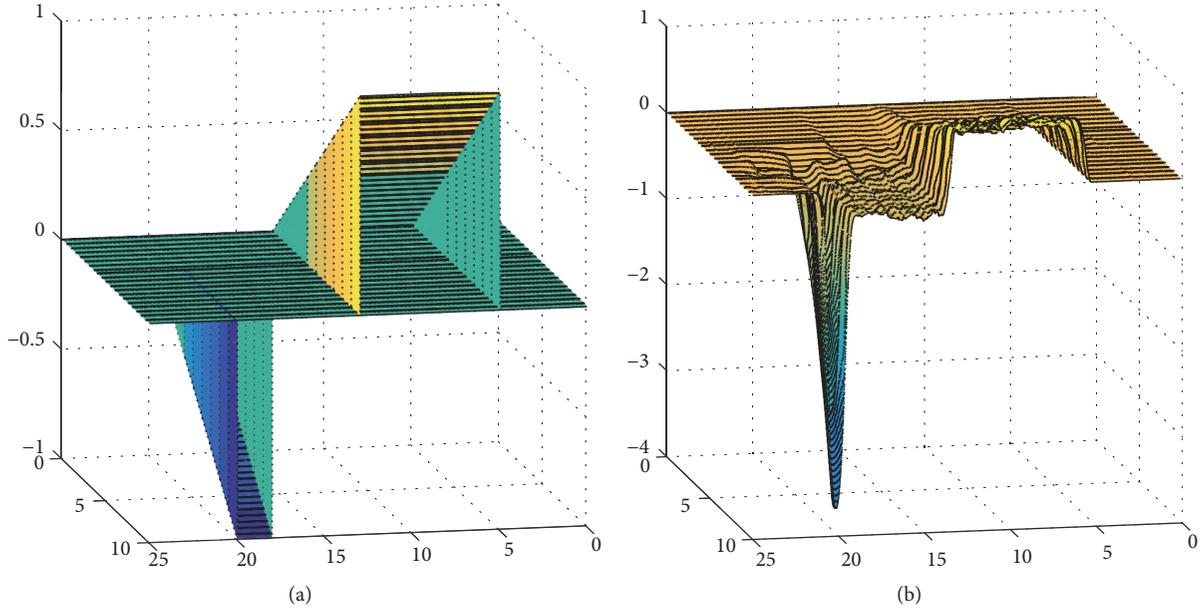


FIGURE 4: (a) Signal control references from on-board computer to longitudinal control actuators and (b) longitudinal acceleration in real test platform in 10 different tests.

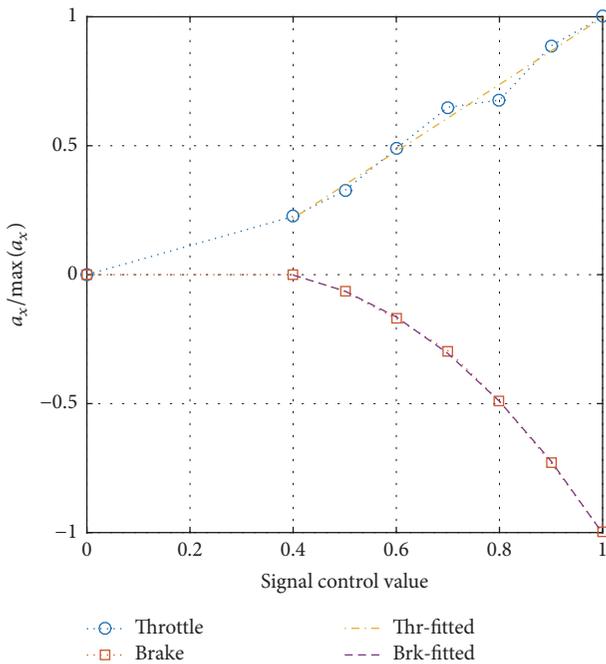


FIGURE 5: Longitudinal behavior of real test platform.

Vehicle speed depends on both accelerator and brake. However, since the dynamics of both actuators present considerable differences (as discussed in Section 2), an individual controller for each mechanism is needed. Figure 6 describes the control block within the automated driving framework. Accelerator and brake controllers are never active at the same time: they are switched depending on the sign of the speed error. Additionally, a lateral controller based on fuzzy

logic [22] has been previously adjusted in order to test the longitudinal algorithms along a specific route.

3.1. PID. Proportional-Integral-Derivative controller, known as PID, is a widely used technique in feedback control applications. According to [23], around 95% of the implemented control loop processes are based on this technique. PID output depends on an input error signal and three tunable gains changing the response of the system. A major advantage is that it reduces the steady state error through the integral action; in addition, derivative gain allows predicting future actions. Equation (3) shows the general formula for a discrete time PID controller using the forward Euler method.

$$u(z) = K_p + K_i T_s \frac{1}{z-1} + \frac{K_d(z-1)}{T_s z}, \quad (3)$$

where $u(z)$ is the control output and K_p , K_i , K_d represent the proportional, integral, and derivative gains, respectively; T_s corresponds to system sample time (1 ms in simulation and 10 ms in real platform). PID controllers have been used for different automated driving longitudinal applications and appear as a practical solution for CC applications where control signals can be modified according to the speed error and selecting the proper gains. Table 3 shows the PID parameters used in this work. These values were tuned in the simulator environment and then tested in the real vehicle.

3.2. ANFIS. Integrating human experience into automated driving algorithms is an idea that has received great interest over the last years. A commonly used technique is fuzzy inference system (FIS), which offers a solution for including human knowledge into controllers design, through a set of membership functions and linguistic if-then rules. Fuzzy controllers have proven its usefulness in different automation

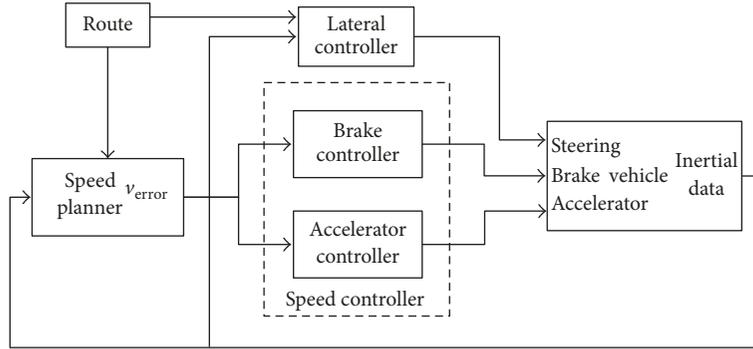


FIGURE 6: Decision and control block of general automated driving framework.

TABLE 3: PID gains.

| | K_p | K_i | K_d |
|-------------|-------|-------|-------|
| Accelerator | 0.5 | 0.005 | 0 |
| Brake | 0.15 | 0.01 | 0.05 |

use cases, such as lateral control [24] and Adaptive Cruise Control (ACC) [25].

However, manual translation of human knowledge into a fuzzy logic database can be a daunting task, and it is complex to know if the designed membership functions and rules are the best choice. This drawback is addressed by ANFIS, combining Takagi and Sugeno fuzzy systems [26] with the learning capacity of adaptive neural network based on human-trained data.

This work implements a two-input-one-output neuro-fuzzy controller designed to depend on two kinds of speed error, absolute (s_{ea}) and relative (s_{er}), as shown in the following equations:

$$\begin{aligned}
 s_{ea} &= s_{ref} - s, \\
 s_{er} &= \frac{s_{ea}}{\Delta ref} \cdot 100, \\
 \Delta ref &= \begin{cases} s_{ref}(i) - s_{ref}(i-1), & \text{if } s_{ref}(i) \neq s_{ref}(i-1) \\ s_{ref}(i), & \text{otherwise,} \end{cases}
 \end{aligned} \quad (4)$$

where s_{ref} is the speed reference and Δref is the step size in a reference change (in Figure 9, Δref values are 15, -5, -5, 5, 5, -7, and -8 [Km/h], resp.). Adding relative error aims to capture the behavior over pedals for the same absolute error, depending on whether it occurs when acceleration is started, maintained, or finished. This strategy for adjusting speed to a reference resembles human-centered driving style at low speeds: pedal is pressed softly at the beginning, then held, and progressively relieved when getting close to reference.

The data for training the ANFIS longitudinal speed controller was gathered from a series of tests done in the simulator. The scenario consisted of a 1 Km highway with a lateral controller adjusted to maintain the vehicle in the center line. Then, USB pedals for brake and accelerator were used to drive the virtual vehicle while maintaining a reference

TABLE 4: ANFIS parameters.

| | Accelerator | Brake |
|--------------------|---------------|---------------|
| Number of inputs | 2 | 2 |
| Number of outputs | 1 | 1 |
| Training algorithm | Hybrid | Hybrid |
| Training data pair | 118994 | 74912 |
| Input type | Gaussian bell | Gaussian bell |
| Output type | Linear | Constant |

TABLE 5: ANFIS output parameters.

| Abs/Rel | Accelerator | | | Brake | | |
|---------|-------------|-------|-------|-------|-------|-------|
| | None | Low | High | None | Low | High |
| None | 0.25 | -0.03 | 0.00 | -0.52 | -0.40 | -0.08 |
| Low | 0.01 | 0.00 | -0.02 | -0.40 | -0.54 | 0.00 |
| High | 0.15 | 0.94 | 2.37 | -0.45 | -1.24 | 0.00 |

speed shown on the screen. For the accelerator controller six tests were performed; the reference was increased in steps of 5, 10, 15, 20, 25, and 30 [Km/h]. Same strategy was used for training the brake, starting at a maximum speed of 30 [Km/h] and repeating the same six trials but decreasing the speed either.

The network structure is composed of two inputs described through Gaussian membership functions and three descriptors (No Error, Low Error, and High Error) in units of [Km/h]. The output of the system is a normalized control action in $[-1; 1]$, where positive values correspond to accelerator controller and negative values are assigned to the brake. Additional information on the neurofuzzy controller is shown in Tables 4 and 5. The former indicates the output values of the controller based on the inputs description. As a basic design rule, both controllers cannot be activated at the same time. Additionally, the controller was trained using a hybrid algorithm [27].

A representation of the control surface for both accelerator and brake is depicted in Figures 7(a) and 7(b), respectively, proving that outputs are within the limits of the values accepted by the actuators. Moreover, control outputs show being smooth and without sudden changes, guaranteeing that the controller will not output sharp commands.

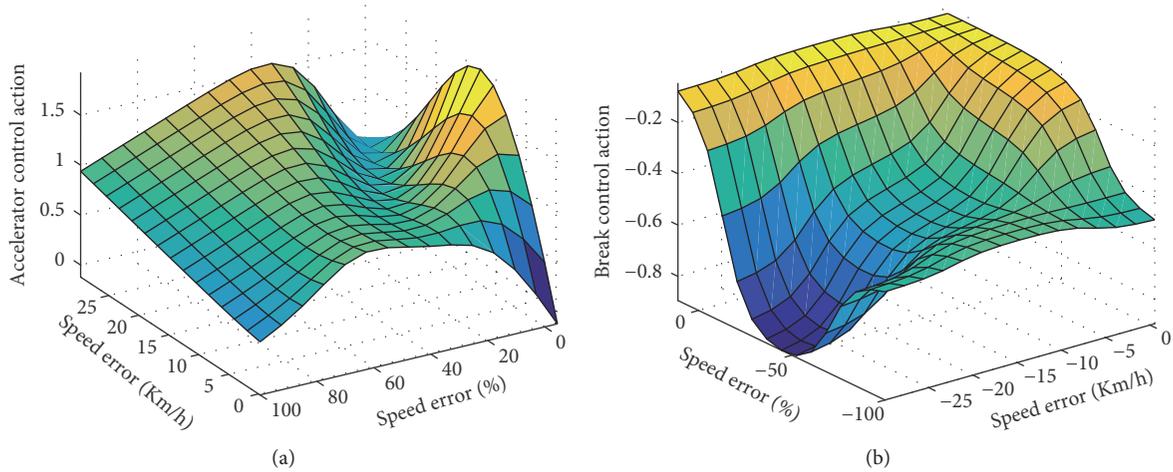


FIGURE 7: (a) Control surface of ANFIS for accelerator; (b) control surface of ANFIS for brake.

The aforementioned algorithm has been designed and trained with data obtained from the simulator, based on the vehicle model explained in Section 3. This responds to the objective of validating the simulation platform as a tool for ADAS design. The benefits include a lower dependence on real vehicle availability and a considerable reduction in trial times.

3.3. MPC. The third algorithm is a model based controller. Model Predictive Control (MPC) [28], also known as receding horizon control, is a concept based on optimal control theory where a plant model and a set of expected inputs are used to predict future system states. It considers a constrained response of the plant and control inputs, optimizing the control values based on an objective function. There are different methods for designing the MPC. For the purpose of this work a real time formulation of a linear model is used.

This method has been largely used in the last decades for applications including high level decision making module for motion planning, along with a secondary feedback control loop [29]; other approaches use open-loop control as a feedforward strategy. One of the major drawbacks of MPC is that it requires a plant model that can describe the real system very accurately. This problem affects especially open-loop solutions. By contrast, feedback based formulations can compensate model errors by updating the states each time the optimization problem is solved.

Next section explains the design of an MPC for CC using a high fidelity vehicle dynamics simulator. Problem formulation and a brief description of the optimization solver are given.

3.3.1. MPC Problem Formulation. The proposed kinematic linear MPC is based on a triple integration chain constructed under three requirements: (i) ability to track reference speed smoothly with low overshoots, (ii) generating acceleration commands feasible for the target vehicle, and (iii) considering the comfort of the passengers. In this sense, the equations that describe the plant are

$$\begin{aligned} \dot{d}_{\text{lon}} &= v_{\text{lon}}, \\ \dot{v}_{\text{lon}} &= a_{\text{lon}}, \\ \dot{a}_{\text{lon}} &= J_{\text{lon}}. \end{aligned} \quad (5)$$

And those can be represented on the classical state-space description, as

$$\begin{bmatrix} \dot{d}_{\text{lon}} \\ \dot{v}_{\text{lon}} \\ \dot{a}_{\text{lon}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{\text{lon}} \\ v_{\text{lon}} \\ a_{\text{lon}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} J_{\text{lon}}. \quad (6)$$

d_{lon} , v_{lon} , a_{lon} , J_{lon} are the longitudinal distance, velocity, acceleration, and jerk, respectively. In this formulation the control variable is jerk, although requirements and constraints in this work are related to future acceleration values. Additionally, the constraints given by the three requirements are

$$\begin{aligned} \min \{v_i, v_{\text{ref}} - 0.5\} &\leq \mathbf{v}(t+k | t) \leq v_{\text{ref}} + 0.5, \\ -a_{\text{dec}} &\leq \mathbf{a}(t+k) \leq a_{\text{acc}}, \\ -j_{\text{dec}} &\leq \mathbf{j}(t+k | t) \leq j_{\text{acc}}, \end{aligned} \quad (7)$$

where $k = k_s t_s$ for $k_s = 0, 1, 2, \dots, N_s$ and t_s is the sample time and N_s is the number of samples used for the MPC prediction horizon. Additionally, v_i is the current speed of the vehicle, v_{ref} is the reference speed, and the factor 0.5 [m/s] is used to limit the overshoot in controller response. Maximum vehicle deceleration is represented by a_{dec} , while a_{acc} is the magnitude of the maximum acceleration; both parameters are related to the dynamic requirements of the vehicle. Lastly, j_{dec} is the maximum deceleration jerk and j_{acc} is the maximum acceleration jerk, which are set according to a passenger comfort criterion as explained in [31].

The objective function used in the optimization problem is the least squared error of the speed:

$$J = \min \left\{ \int_{t_i}^{t_i+t_H} (v - v_{\text{ref}})^2 dt \right\}, \quad (8)$$

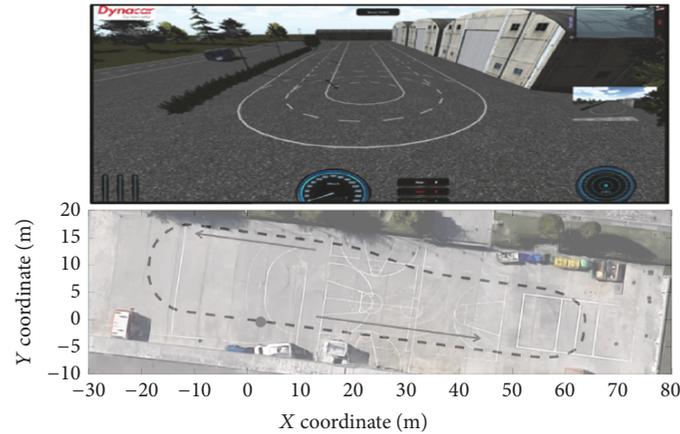


FIGURE 8: Derio 700 circuit: simulation and real one [30].

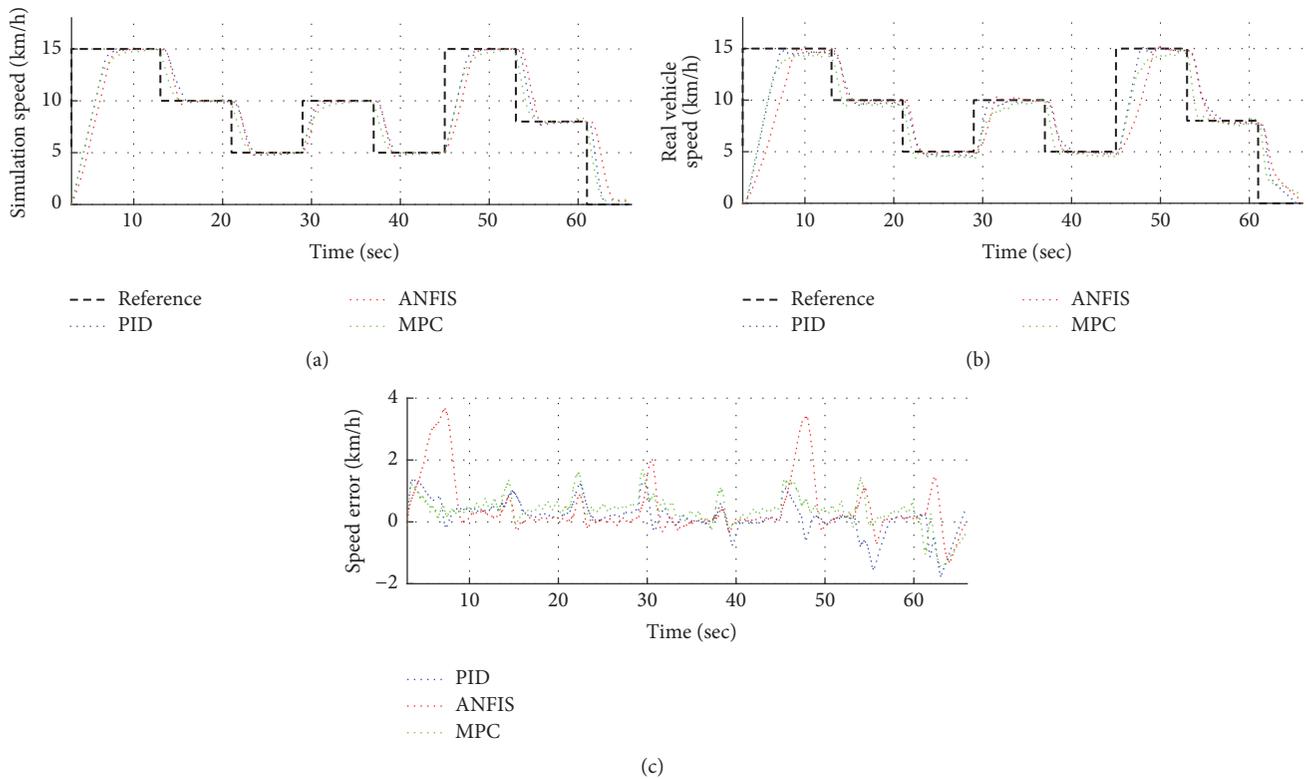


FIGURE 9: Speed response in simulation and real vehicle.

where t_i is the time at which control actions are optimized, and t_H is the prediction horizon; that is, $t_H = t_s \cdot N_s$. In addition, Table 6 summarizes the parameters used in the optimization problem for the MPC.

3.3.2. MPC Solver. The solver used to calculate the solution of the MPC is the ACADO toolkit [32]. It is an open-source and self-contained library based on C++ code, designed to solve linear and nonlinear dynamic problems under multiobjective optimization functions. Moreover, ACADO toolkit functionality supports interfacing with Matlab/Simulink environment [33], which makes it an ideal

option for integration with the presented automated driving framework.

4. Test and Results

An evaluation of three CC algorithms is presented, based on a comparative analysis between controllers in both simulation and real vehicle. First, speed response is analyzed using the tracking error as a performance indicator. Additionally, mean and maximum absolute acceleration are discussed.

Experimental tests were carried out within the facilities of Tecnalia Research & Innovation, in a closed circuit of 180

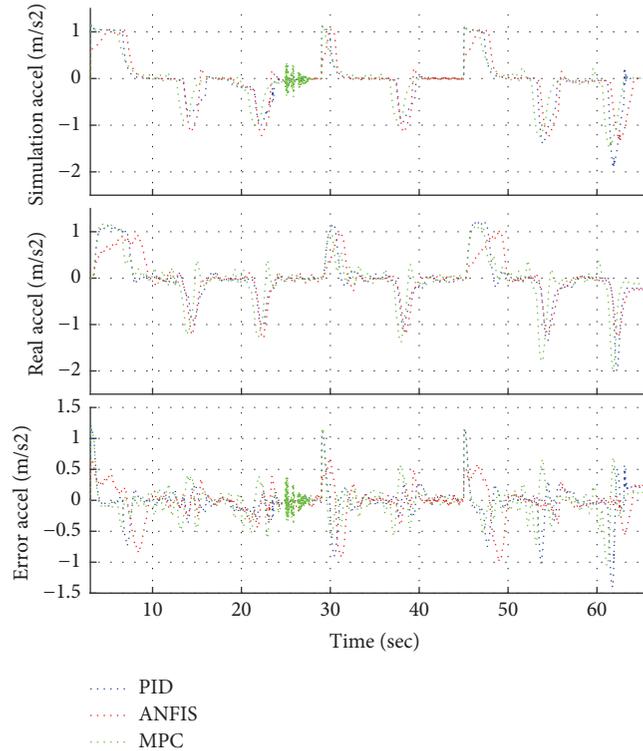


FIGURE 10: Acceleration response in simulation and real vehicle.

TABLE 6: MPC parameters.

| Parameter | Value |
|-------------------------------|---------|
| v_i [m/s] | Dynamic |
| v_{ref} [m/s] | Dynamic |
| a_{dec} [m/s ²] | 3.15 |
| a_{acc} [m/s ²] | 1.15 |
| j_{dec} [m/s ³] | 2 |
| j_{acc} [m/s ³] | 2 |
| t_s [s] | 0.5 |
| N_s | 10 |

[m] length. It is comprised of a straight path, two turns, and one lane change. Same scenario was replicated in the virtual environment as shown in Figure 8. Vehicle maximum speed was set to 15 [Km/h] with multiple reference changes delivered by the speed planner.

4.1. Speed Control Analysis. The speed response of the three controllers is depicted in Figure 9. It includes results from simulation and experimental tests, along with the error between both platforms. All controllers stick to the reference, showing similar response in simulated and real vehicle. The PID shows the faster settling time for the three acceleration maneuvers ($t \in [3-13] \cup [29-37] \cup [45-53]$). Meanwhile, MPC is the slower in reaching the steady state when accelerating; however, it is the faster in braking maneuvers. Neurofuzzy controller is the less aggressive; nonetheless, far from being a disadvantage, it is a good indicator of acceleration comfort.

Moreover, when analyzing the tracking error once the controller has reached 95% of the reference, the neurofuzzy technique shows smaller tracking error.

Previous observations can be complemented with Table 7, using average, median, and RMS of the tracking errors. In simulation, the best controller is the MPC with an overall average error of 1.27 [Km/h], followed by the PID and the ANFIS. Same order is obtained when evaluating this variable in the real vehicle. An important result is that analyzing median and RMS, the order is the same in the two platforms. This information suggests that the simulation tool represents the behavior of the real vehicle correctly.

In addition, Figure 9(c) shows the error comparing the controllers in simulation and experimentally. The neurofuzzy controller has the highest error peak; the controllers output plot (Figure 12) explains this behavior: the neurofuzzy system command softens initial acceleration that is progressively increased. This situation makes real vehicle reaction slower, and although the controller shows similar acceleration in both cases (≈ 1 [m/s²]), a delay in the acceleration action produces a temporary error peak. Moreover, the average, mean, and RMS errors are below 0.5 [Km/h] for all the controllers; based on smaller reference change (5 [Km/h]), it represents an approximated overall error of 10% between the simulated and the real vehicle.

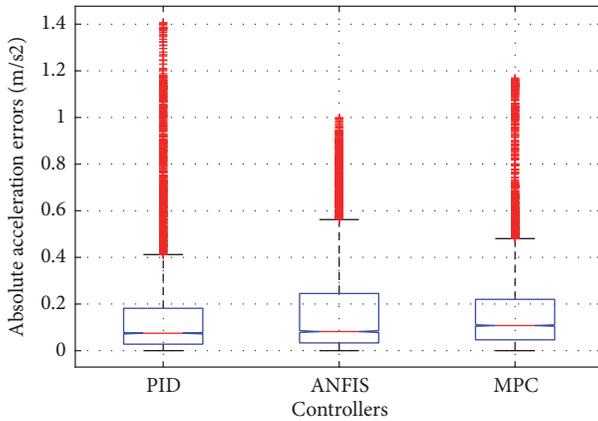
4.2. Acceleration Response Analysis. Acceleration performance is evaluated based on Figure 10. The test around the circuit produced 8 acceleration peaks, one per reference change, with a similar response in both simulated and real platforms. Results are consistent with the conclusions

TABLE 7: Tracking errors [Km/h].

| | Track error avg | | | Track error median | | | Track error RMS | | |
|-------|-----------------|------|------|--------------------|------|------|-----------------|------|------|
| | Sim | Real | Diff | Sim | Real | Diff | Sim | Real | Diff |
| PID | 1.45 | 1.65 | 0.20 | 0.16 | 0.36 | 0.2 | 3.04 | 3.28 | 0.24 |
| ANFIS | 1.70 | 2.05 | 0.35 | 0.07 | 0.18 | 0.11 | 3.49 | 3.93 | 0.44 |
| MPC | 1.27 | 1.56 | 0.29 | 0.19 | 0.50 | 0.31 | 2.80 | 3.08 | 0.28 |

TABLE 8: Acceleration errors [m/s^2].

| | Acceleration avg | | | Acceleration median | | | Acceleration RMS | | |
|-------|------------------|-------|-------|---------------------|-------|-------|------------------|-------|-------|
| | Sim | Real | Error | Sim | Real | Error | Sim | Real | Error |
| PID | 0.279 | 0.276 | 0.003 | 0.038 | 0.070 | 0.032 | 0.502 | 0.480 | 0.022 |
| ANFIS | 0.282 | 0.273 | 0.009 | 0.049 | 0.070 | 0.021 | 0.486 | 0.442 | 0.044 |
| MPC | 0.278 | 0.286 | 0.08 | 0.083 | 0.100 | 0.017 | 0.471 | 0.495 | 0.024 |

FIGURE 11: Boxplot of absolute acceleration errors between simulation and real platform [m/s^2].

extracted in the speed analysis: ANFIS controller softness can be observed while both PID and MPC techniques produce faster reactions.

Table 8 shows a small error when comparing the mean of the absolute acceleration values in the two platforms (below $0.1 [m/s^2]$). Median and RMS results present errors below $0.05 [m/s^2]$. These results are combined with the quartile analysis in Figure 11, showing that the difference between real and simulated vehicle acceleration is smaller than $0.2 [m/s^2]$ of 75% of the time. Red markers represent outlier values (around 9% of the collected data); most of them are caused by the variable delay between the simulated signals and the real ones.

Results discussed above can be extended by considering the errors during the eight zones of peak acceleration (Table 9). Data shows an average error of $0.09 [m/s^2]$ for the PID controller, $0.07 [m/s^2]$ corresponding to the ANFIS technique, and $0.28 [m/s^2]$ for the MPC. In conclusion, the neurofuzzy controller presents smaller values for acceleration averages and errors between the simulated and the real vehicle. Moreover, overall errors between both vehicles are below 10% based on a maximum absolute acceleration of

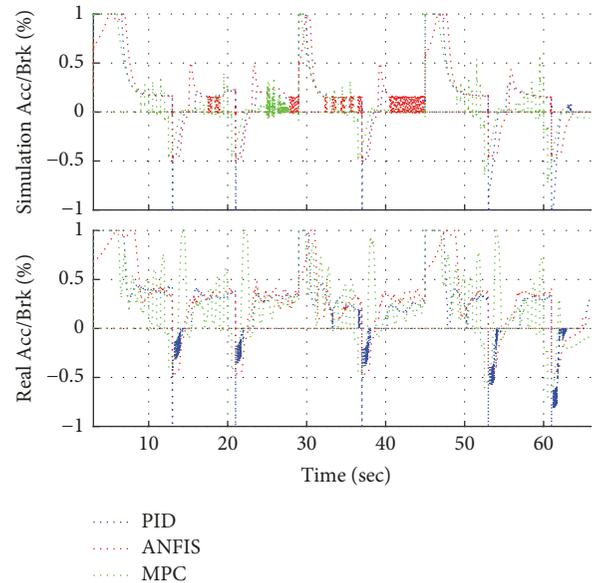


FIGURE 12: Controller output: simulation and real one.

$2 [m/s^2]$ and considering that most of the absolute acceleration errors (close to 75%) are below $0.2 [m/s^2]$ (see Figure 11). This is consistent with the speed analysis presented before.

5. Conclusion and Future Work

This work has presented the development and implementation of three longitudinal controllers for CC at low speed. The algorithms were designed in an in-house simulation framework based on a multibody vehicle model. Then, controllers were applied in an instrumented Renault Twizy. Software implementation was based on a general automated driving architecture developed in Matlab/Simulink.

The major contributions of this work include the following:

- (i) A complete framework for automated driving is presented and implemented in virtual and real vehicles.

TABLE 9: Peaks of acceleration errors [m/s²].

| | PID | | | ANFIS | | | MPC | | |
|--------|------|------|-------|-------|------|-------|------|------|-------|
| | Sim | Real | Error | Sim | Real | Error | Sim | Real | Error |
| Step 1 | 1.20 | 1.14 | 0.06 | 1.05 | 0.92 | 0.13 | 1.17 | 1.16 | 0.01 |
| Step 2 | 0.80 | 0.92 | 0.12 | 1.14 | 1.22 | 0.08 | 0.91 | 1.22 | 0.31 |
| Step 3 | 0.99 | 1.03 | 0.04 | 1.22 | 1.28 | 0.06 | 1.00 | 1.26 | 0.26 |
| Step 4 | 1.13 | 1.16 | 0.03 | 1.06 | 0.96 | 0.10 | 1.14 | 1.04 | 0.1 |
| Step 5 | 0.94 | 1.17 | 0.23 | 1.15 | 1.21 | 0.06 | 0.92 | 1.38 | 0.54 |
| Step 6 | 1.14 | 1.23 | 0.09 | 1.05 | 1.05 | 0.00 | 1.13 | 1.12 | 0.01 |
| Step 7 | 1.40 | 1.34 | 0.06 | 1.25 | 1.20 | 0.05 | 1.25 | 1.79 | 0.54 |
| Step 8 | 1.90 | 2.00 | 0.1 | 1.32 | 1.27 | 0.05 | 1.48 | 1.98 | 0.50 |

- (ii) The framework is validated as a tool for simulation-based design of ADAS functionality. Experiments show that the results obtained in the simulation tool are representative of its behavior in the real vehicle within an approximated error of 10% between platforms. The benefits include reducing the time and cost of validating automated driving algorithms while increasing performance.
- (iii) A use case of longitudinal CC based on three different control techniques is presented. The use case covers the full development process: characterization of the real vehicle, adjusting vehicle parameters in the simulator, design and implementation of desired functionality, testing and tuning in simulation platform, and validation in real vehicle.

These results open new possibilities for developing automated driving algorithms using this methodology, including new scenarios, other use cases, and more complex algorithms. Future works include the validation of the simulation framework for lateral maneuvers, such as overtaking and lane keeping. Design of controllers for cooperative maneuvers will be considered as well.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the ESCAL Project ENABLE-S3 (with Grant no. 692455-2) for the support in the development of this work.

References

- [1] Graham Cookson, "Europes Traffic Hotspots Measuring the impact of congestion," INRIX Research, 2016.
- [2] Connectivity and Automated Driving, "Automated Driving Roadmap," ERTRAC Working Group, 2017.
- [3] On-Road Automated Vehicle Standards Committee SAE, "Taxonomy and definitions for terms related to onroad motor vehicle automated driving systems," 2014.
- [4] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2014.
- [5] X. Jin, G. Yin, and J. Wang, "Robust fuzzy control for vehicle lateral dynamic stability via Takagi-Sugeno fuzzy approach," in *Proceedings of the 2017 American Control Conference, ACC 2017*, pp. 5574–5579, Seattle, WA, USA, May 2017.
- [6] Benjamin Englander et al., "Collision Avoidance System for Buses, Managing Pedestrian Detection and Alerts Near Bus Stops," in *Proceedings of the 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, Detroit, Michigan, USA, 2017.
- [7] Mariano Sans, "Semi-automatic method for assisting with parking a motor vehicle and associated device," US Patent 9,731,715. 2017.
- [8] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.
- [9] M. James Anderson, *Autonomous Vehicle Technology: A Guide for Policymakers*, Rand Corporation, Santa Monica, California, USA, 2014.
- [10] Waymo, "Waymo Safety Report On The Road to Fully Self-Drivin," Safety Report, 2017.
- [11] F. Bounini, D. Gingras, V. Lapointe, and H. Pollart, "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking," in *Proceedings of the 12th IEEE Vehicle Power and Propulsion Conference, VPPC 2015*, Canada, October 2015.
- [12] S. Prialé Olivares, N. Rebernik, A. Eichberger, and E. Stadlober, "Virtual Stochastic Testing of Advanced Driver Assistance Systems," in *Advanced Microsystems for Automotive Applications 2015*, Lecture Notes in Mobility, pp. 25–35, Springer, Cham, Switzerland, 2016.
- [13] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Development of a drive-in driver-in-the-loop fully immersive driving simulator for virtual validation of automotive systems," in *Proceedings of the 81st IEEE Vehicular Technology Conference, VTC Spring 2015*, Glasgow, UK, May 2015.
- [14] V. Milanés, J. Villagrà, J. Pérez, and C. González, "Low-speed longitudinal controllers for mass-produced cars: A comparative study," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 620–628, 2012.
- [15] P. Ioannou and Z. Xu, "Throttle and brake control systems for automatic vehicle following," *Journal of Intelligent Transportation Systems*, vol. 1, no. 4, pp. 345–377, 1994.

- [16] David González and Joshué Pérez, "Control architecture for Cybernetic Transportation Systems in urban environments," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 1119–1124, Australia, June 2013.
- [17] R. Lattarulo, J. Pérez, and M. Dendaluze, "A complete framework for developing and testing automated driving controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 258–263, 2017.
- [18] Joshué Pérez et al., "Trajectory generator for autonomous vehicles in urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 409–414, Karlsruhe, Germany, May 2013.
- [19] J. Cuadrado, D. Dopico, M. A. Naya, and M. Gonzalez, "Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators," *Multibody System Dynamics*, vol. 12, no. 2, pp. 117–132, 2004.
- [20] J. Cuadrado, D. Vilela, I. Iglesias, A. Martín, and A. Peña, "A multibody model to assess the effect of automotive motor in-wheel configuration on vehicle stability and comfort," in *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*, pp. 1083–1092, 2013.
- [21] I. Iglesias-Aguinaga, A. Martín-Sandi, and A. Pena-Rodríguez, "Vehicle modelling for real time systems application. The virtual rolling chassis," *Dyna (Spain)*, vol. 88, no. 2, pp. 206–215, 2013.
- [22] J. Pérez, V. Milanés, and E. Onieva, "Cascade architecture for lateral control in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 73–82, 2011.
- [23] Karl Johan Åström and Tore Hägglund, *Advanced PID Control*, ISA-The Instrumentation, Systems and Automation Society, Research Triangle Park, North Carolina, USA, 2006.
- [24] X. Wang, M. Fu, H. Ma, and Y. Yang, "Lateral control of autonomous vehicles based on fuzzy logic," *Control Engineering Practice*, vol. 34, pp. 1–17, 2015.
- [25] J. E. Naranjo, C. González, R. García, T. De Pedro, and M. A. Sotelo, "Using fuzzy logic in automated vehicle control," *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 36–45, 2007.
- [26] T. Takagi and M. Sugeno, "Derivation of Fuzzy Control Rules from Human Operator's Control Actions," *IFAC Proceedings Volumes*, vol. 16, no. 13, pp. 55–60, 1983.
- [27] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [28] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2–4, pp. 265–291, 2005.
- [29] H. Jiang, Z. Wang, Q. Chen, and J. Zhu, "Obstacle avoidance of autonomous vehicles with CQP-based model predictive control," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 1668–1673, Hungary, October 2016.
- [30] Ray Lattarulo et al., "A Speed Planner Approach Based On Bzier Curves Using Vehicle Dynamic Constrains and Passengers Comfort," in *in the Proceedings of the The International Symposium on Circuits and Systems*, Florence, Italy, 2018.
- [31] Q. Huang and H. Wang, "Fundamental study of jerk: Evaluation of shift quality and ride comfort," *SAE Technical Papers*, 2004.
- [32] B. Houska et al., "ACADO Toolkit Users Manual," <http://www.acadotoolkit.org>. 2009–2013.
- [33] D. Ariens, B. Houska, and H. J. Ferreau, "ACADO for Matlab Users Manual," <http://www.acadotoolkit.org>. 2010–2011.



Hindawi

Submit your manuscripts at
www.hindawi.com

