

# Optimal LPV-based Control and Estimation for Autonomous Vehicles

Eugenio Alcalá  
CS2AC  
UPC  
Barcelona, Spain  
eugenio.alcala@upc.edu

Marc Facerías  
CS2AC  
UPC  
Barcelona, Spain  
marc.facierias@estudiant.upc.edu

Vicenç Puig  
CS2AC  
UPC  
Barcelona, Spain  
vicenc.puig@upc.edu

**Abstract**—This article presents an approach to address the problem of designing advanced control and estimation techniques for the autonomous driving guidance. In particular, this work takes advantage of the properties of polytopic LPV systems and predictive optimal control to guide the vehicle along a planned trajectory. Linear Parameter Varying (LPV) theory is used to model the dynamics of the vehicle and implement an LPV-Model Predictive Controller (LPV-MPC) that can be computed online with reduced computational cost. Furthermore, the LPV framework is used to design an optimal observer that estimates vehicle variables that cannot be measured. The control and estimation scheme is validated in simulation using the Robotic Operating System (ROS) framework where its effectiveness is demonstrated.

**Index Terms**—Autonomous driving, MPC control, LPV modeling, optimal estimation

## I. INTRODUCTION

In the last few years, we have witnessed a fast technology advance in the automotive area to improve traffic mobility and city congestion but mostly to guarantee safer roads. Currently, we can find a wide variety of research lines in the automotive field offering solutions to the up-coming driving challenges from an autonomous point of view. For example, in [1], the authors present a lane keeping control algorithm based on differential steering for guiding the vehicle autonomously. Also, in [2], a steering control algorithm is proposed for accurate, smooth, and computationally inexpensive control of autonomous vehicles. This type of approaches work appropriately for the mentioned applications. However, most of these algorithms perform lateral and longitudinal control separately or even using poor representations of the vehicle dynamics which sometimes may lead to undesirable behaviours.

Nowadays, new challenges appear on the near horizon which require more sophisticated techniques to guarantee certain levels of driving safety. Hence, handling difficult driving situations guaranteeing safety requires advanced control and estimation techniques. Generally, these algorithms require of high computational speed, coupled

dynamics and accurate vehicle models to carry their performance to a high standard.

From the control perspective, one of the most interesting techniques in autonomous driving is the MPC. This allows to define a constrained problem that is solved optimally taking into account the future evolution of a vehicle model. It is common to find in the literature works solving the path tracking problem using the mentioned control strategy [3]. In [4]–[6], different ways of dealing with the MPC strategy using an LPV representation are addressed. The advantage of the LPV modeling approach is that the non-linear model can be expressed as a combination of linear models with parameter varying with some scheduling variables without using linearization [7].

The state measurement task will depend on the type of sensors installed on the vehicle and will be of vital importance for the application of certain control strategies. Owing to the difficulty of measuring some basic vehicle variables, the state estimation problem is of great interest in the automotive field. A brief review of the most used family of estimators in this application is made in [8].

This paper presents a control-estimation architecture for solving the autonomous driving problem in urban environments. This solution is based on two layers. The highest one deals with the planning, high level control and the estimation tasks. The lowest one deals with the low level control using PID controllers and is in charge of managing some security elements. Both high level control and estimation are designed using optimal control theory. The proposed scheme is integrated with a trajectory generation module and tested in a simulated scenario using ROS.

This paper is structured as follows: Section 2 presents and describes the different types of vehicle models used for control and estimation purposes. In Section 3, the LPV-MPC control formulation is developed. Section 4 shows the polytopic estimation law using the LPV-LQR approach. Section 5 presents the implementation and results. Finally,

This work has been funded by the Spanish Ministry of Economy and Competitiveness (MINECO) and FEDER through the projects SCAV (ref. DPI2017-88403-R) and HARCRICS (ref. DPI2014-58104-R). The author is supported by a FI AGAUR grant (ref 2017 FI B00433).

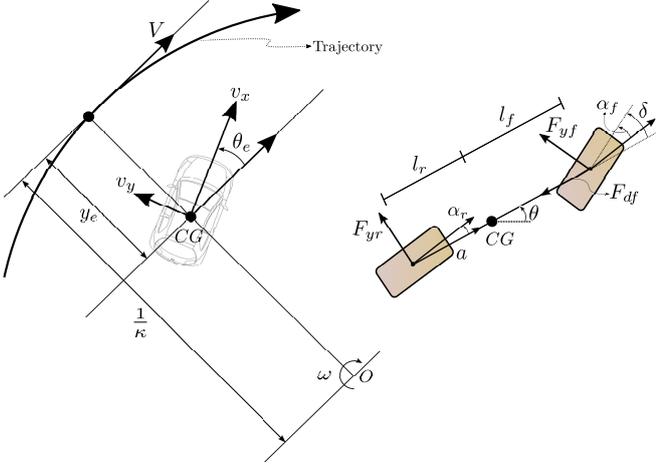


Fig. 1. Variables representation. Left: error variables representation. Right: dynamic model variables

Section 6 presents the conclusions of the work.

## II. TESTING VEHICLE & MODELING

The vehicle in question used for validating both the controller and estimator is a Kia Niro. This has been physically modified to operate autonomously by applying actions to its actuators, i.e. steering wheel, brake pump and throttle valve. The vehicle counts with a complete net of sensors for performing localization and perception tasks such as stereo cameras, RADAR, LIDAR and GPS+IMU.

Model-based control and estimation techniques require a mathematical formulation of the system and, in our case, in a state space formulation. In this section, simulation, control and estimation models are presented. First, the non-linear model used for simulation is presented and then the LPV representations for control and estimation are formulated for obtaining the models that will be used later for control and estimation.

### A. Non-linear vehicle model

This car model is represented using the bicycle model approach where the tire model produces the set of lateral forces as a linear relation with the slip angles. The model parameters have been adjusted following an experimental data-based identification. Then, denoting the state and control vectors, respectively, as

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ y_e \\ \theta_e \end{bmatrix}, \quad u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \quad (1a)$$

the continuous time non-linear model is presented as

$$\dot{x} = f(x, u, \kappa), \quad (1b)$$

where variable  $\kappa$  represents the curvature. The model (1b) is governed by the following equations

$$\begin{aligned} \dot{v}_x &= a + \frac{-F_{yf} \sin \delta - F_{df}}{m} + \omega v_y \\ \dot{v}_y &= \frac{F_{yf} \cos \delta + F_{yr}}{m} - \omega v_x \\ \dot{\omega} &= \frac{F_{yf} l_f \cos \delta - F_{yr} l_r}{I} \\ \dot{y}_e &= v_x \sin \theta_e + v_y \cos \theta_e \\ \dot{\theta}_e &= \omega - \frac{v_x \cos \theta_e - v_y \sin \theta_e}{1 - y_e \kappa} \kappa \\ \alpha_f &= \delta - \tan^{-1} \left( \frac{v_y}{v_x} - \frac{l_f \omega}{v_x} \right) \\ \alpha_r &= -\tan^{-1} \left( \frac{v_y}{v_x} + \frac{l_r \omega}{v_x} \right), \end{aligned} \quad (1c)$$

where

$$\begin{aligned} F_{yf} &= C_f \alpha_f \\ F_{yr} &= C_r \alpha_r \\ F_{df} &= \mu mg + \frac{1}{2} \rho C_{dA} v_x^2. \end{aligned} \quad (1d)$$

State variables  $v_x$ ,  $v_y$  and  $\omega$  represent the body frame velocities, i.e. linear in  $x$ , linear in  $y$  and angular velocities, respectively. States  $y_e$  and  $\theta_e$  represent the body frame lateral error and the orientation error, respectively (see Figure 1). The control variables  $\delta$  and  $a$  are the steering angle at the front wheels and the longitudinal acceleration vector on the rear wheels, respectively.  $F_{yf}$  and  $F_{yr}$  are the lateral forces produced in front and rear tires, respectively. Front and rear slip angles are represented as  $\alpha_f$  and  $\alpha_r$ , respectively, and  $C_f$  and  $C_r$  are the front and rear tire stiffness coefficients.  $m$  and  $I$  represent the vehicle mass and inertia and  $l_f$  and  $l_r$  are the distances from the vehicle center of mass to the front and rear wheel axes, respectively.  $F_{df}$  is the composition of static friction force and drag force that acts opposing the movement of the vehicle.  $\mu$ ,  $\rho$  and  $g$  are the static friction coefficient, the air density at 25°C and the gravity, respectively.  $C_{dA}$  is the product of drag coefficient and vehicle cross sectional area. All the dynamic vehicle parameters are properly defined in Table I.

TABLE I  
DYNAMIC MODEL PARAMETERS OF THE VEHICLE

Parameter	Value	Parameter	Value
$l_f$	1.52 m	$l_r$	1.22 m
$m$	1554 kg	$I$	2200 kg m <sup>2</sup>
$C_f$	59054	$C_r$	107123
$\mu$	0.2	$\rho$	1.225 kg m <sup>3</sup>
$C_{dA}$	1.64	$g$	9.81 $\frac{m}{s^2}$

### B. LPV representation for control

LPV systems are defined as finite-dimensional linear time-varying plants whose state space matrices are fixed functions of some vector of varying and measurable parameters [7]. Obtaining the LPV formulation of a non-linear system may be sometimes a tedious and not trivial task. In this particular

case, trying to obtain an LPV representation of (1) may result on many different options and not all of them with the same quality representation. Before transforming the non-linear model into some of its possible LPV representations, the arctangent function in (1c) is neglected to simplify the formulation. Note that, this assumption is coherent since the slip angle value remains below 0.1 *rad* for normal driving where  $\tan^{-1}(\alpha) \approx \alpha$ . Then, denoting the state and control vectors as in (1a), the non-linear model (1) is transformed into the following continuous LPV representation

$$\dot{x} = A(\zeta)x + B(\zeta)u, \quad (2a)$$

by embedding the non-linearities within varying linear parameters. Each one of these parameters is function of the vector of scheduling variables defined as

$$\zeta := [v_x \quad v_y \quad \theta_e \quad \kappa \quad y_e \quad \delta]^T. \quad (2b)$$

The scheduling variables are system states, inputs and even exogenous references as the case of  $\kappa$ .

Then, the LPV matrices, i.e.  $A(\zeta)$  and  $B(\zeta)$ , are obtained using the non-linear embedding approach [9] as

$$A(\zeta) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & A_{42} & 0 & 0 & A_{45} \\ -A_{51} & A_{52} & 1 & 0 & 0 \end{bmatrix}, \quad (2c)$$

$$B(\zeta) = \begin{bmatrix} -\frac{1}{m} \sin \delta C_f & 1 \\ \frac{1}{m} \cos \delta C_f & 0 \\ \frac{1}{I} \cos \delta C_f l_f & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (2d)$$

being

$$\begin{aligned} A_{11} &= \frac{-\mu g}{v_x} - \frac{\rho C_{dA} v_x}{2m} \\ A_{12} &= \frac{C_f \sin \delta}{m v_x} \\ A_{13} &= \frac{C_f l_f \sin \delta}{m v_x} + v_y \\ A_{22} &= -\frac{C_r + C_f \cos \delta}{m v_x} \\ A_{23} &= -\frac{C_f l_f \cos \delta - C_r l_r}{m v_x} - v_x \\ A_{32} &= -\frac{C_f l_f \cos \delta - l_r C_r}{I v_x} \\ A_{33} &= -\frac{C_f l_f^2 \cos \delta + l_r^2 C_r}{I v_x} \\ A_{51} &= \frac{\kappa}{(1 - y_e \kappa)} \\ A_{52} &= \frac{\kappa}{(1 - y_e \kappa)} \sin \theta_e \\ A_{42} &= \cos \theta_e, \quad A_{45} = v_x. \end{aligned} \quad (2e)$$

### C. LPV representation for estimation

The observation problem implies a correct estimation of the real vehicle variables. Then, denoting the vehicle state and input vectors, respectively, as

$$x = \begin{bmatrix} v_x \\ v_y \\ \omega \\ x_p \\ y_p \\ \theta \end{bmatrix}, \quad u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \quad (3a)$$

where  $x_p$ ,  $y_p$  and  $\theta$  represent the vehicle position and orientation, respectively, the continuous time LPV representation of the system used for observation is

$$\dot{x} = A(\phi)x + Bu, \quad (3b)$$

with

$$A(\phi) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 \\ \cos(\theta) & -\sin(\theta) & 0 & 0 & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (3c)$$

$$B = \begin{bmatrix} 0 & 1 \\ \frac{1}{m} C_f & 0 \\ \frac{1}{I} C_f l_f & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3d)$$

where the vector of scheduling variables is defined as

$$\phi := [v_x \quad v_y \quad \theta \quad \delta]^T \quad (3e)$$

and the embedded parameters in  $A(\phi)$  where defined in (2e).

## III. CONTROL

In this section, we present the control layer using the MPC technique with the LPV model representation (2) of the non-linear vehicle model. The prediction of future vehicle behaviour is handled by using the LPV model within the optimal control. This imply that, at every time instant, an instantiation of the non-linear vehicle model computed with a known scheduling vector is required. Such a vector can be given by the trajectory planner or by the prediction made in the previous iteration. Note that, using the LPV modeling we avoid expensive computations as in linearisation-based approaches as well as the model instantiation remain more precise. This allows to formulate the MPC problem as a quadratic optimization problem that is solved at each time  $k$  to determine the control actions considering that the values of  $x_k$  and  $u_{k-1}$  are known.

$$\begin{aligned} \min_{\Delta U_k} J_k &= \sum_{i=0}^{H_p-1} \left( (r_{k+i} - x_{k+i})^T Q (r_{k+i} - x_{k+i}) \right. \\ &\quad \left. + \Delta u_{k+i} R \Delta u_{k+i} \right) + x_{k+H_p}^T Q x_{k+H_p} \end{aligned}$$

s.t.

$$\begin{aligned} x_{k+i+1} &= x_{k+i} + (A(\zeta_{k+i})x_{k+i} + B(\zeta_{k+i})u_{k+i})T_s \\ u_{k+i} &= u_{k+i-1} + \Delta u_{k+i} \\ \Delta U_k &\in \Delta \Pi \\ U_k &\in \Pi \\ y_e &\in [\bar{y}_e, \underline{y}_e] \\ x_{k+0} &= \hat{x}_k, \end{aligned} \quad (4)$$

where  $x$  and  $u$  were defined in (1a),  $\hat{x}$  is the estimated state vector,  $r = [v_x \quad 0 \quad 0 \quad 0 \quad 0]^T$  is the reference

vector provided by the trajectory planner and  $H_p$  is the prediction horizon. The tuning matrices  $Q \in \mathbb{R}^{5 \times 5}$  and  $R \in \mathbb{R}^{2 \times 2}$ , are semi-positive definite in order to set a convex cost function. The time discretization is carried out using Euler approach and the constant sampling time  $T_s$ . Constant sets  $\Pi = \{u_k | A_u u_k = b_u, u_k \geq 0\}$  and  $\Delta\Pi = \{\Delta u_k | A_{\Delta u} \Delta u_k = b_{\Delta u}, \Delta u_k \geq 0\}$  constraint the system inputs and their variations, respectively.

#### IV. ESTIMATION

Due to the lack of available sensors for measuring all the vehicle states such as the lateral velocity at the vehicle's centre of gravity, the design of a state estimator has been considered in this section (see Fig. 2). Such an estimator has been designed following an optimal approach exploiting the duality between the Linear Quadratic Regulator (LQR) and Kalman filter approaches. Following the design and results showed in [10], this section presents an LPV-LQR observer with both stability and performance criteria integrated in a single design procedure.

The state estimation scheme is also based on LPV gain scheduling techniques

$$\dot{\hat{x}} = (A(\phi) - L(\phi)C)\hat{x} + Bu + L(\phi)y, \quad (5)$$

where  $L(\phi)$  is the observer gain that depends on the vector of scheduling variables  $\phi$  defined in (3e),  $y$  represents the vector of measured variables and  $A(\phi)$  and  $B$  are the system matrices presented in (3). Considering the available sensors, the output matrix is defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where it can be seen that the lateral velocity cannot be measured.

##### A. Description of the design method

To design the observer gain  $L$  in (5), a polytopic LPV approximation of (3) is used where the scheduling vector  $\phi$  ranges over a fixed polytope  $\Theta$ , resulting in the following representation

$$\dot{\hat{x}} = \sum_{i=1}^N \mu_i(\phi) (A_i \hat{x} + B u) \quad (7)$$

where the system matrices  $A_i$  and  $B$  define the so-called *vertex systems*,  $N$  is the number of polytopic vertexes and  $\mu(\cdot)$  is polytopic interpolation function given by

$$\mu_i(\phi) = \prod_{j=1}^N \xi_{ij}(\eta_0^j, \eta_1^j), \quad (8)$$

$$\eta_0^j = \frac{\bar{\phi}_j - \phi_j(t)}{\bar{\phi}_j - \underline{\phi}_j} \quad (9)$$

$$\eta_1^j = 1 - \eta_0^j$$

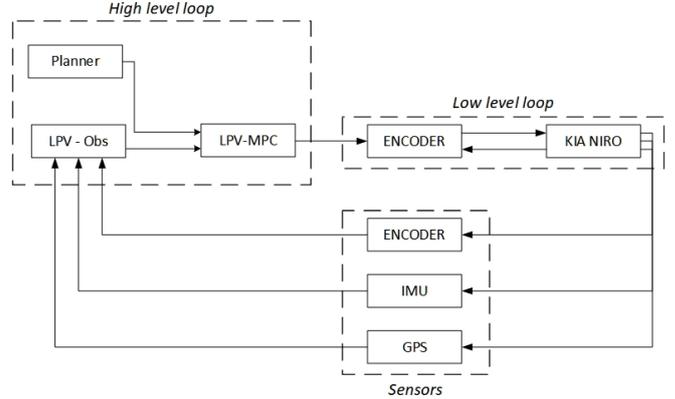


Fig. 2. Autonomous driving scheme divided in two levels

where  $\xi_{ij}(\eta_0^j, \eta_1^j)$  corresponds to any of the weighting function that depend on each rule  $i$ . In addition, next conditions must be satisfied

$$\sum_{i=1}^N \mu_i(\phi) = 1, \quad \mu_i(\phi) \geq 0, \quad \forall \phi \in \Theta. \quad (10)$$

Finally, the observer gain is computed at every instant of time as

$$L(\phi) = \sum_{i=1}^N \mu_i(\phi) L_i \quad (11)$$

where the set of  $L_i$  are obtained using the LQR formulation proposed in [10].

#### V. IMPLEMENTATION

In this section, we are going to present the implementation of the control technique stated in previous pages of this paper, along with its integration with a low-level control loop already installed in the car. This second loop applies the control actions, acceleration and steering, to the system through the pedals and steering wheel, respectively.

##### A. Control set up

The main goal of the controller is being able to follow properly the references provided by the trajectory planner. This planning algorithm has been designed following the strategy presented in [11]. Those references have a low velocity profile while enforcing comfortable behaviour suitable for driving in a residential environment.

As we aim to implement the control scheme in a real car there are several considerations regarding temporal correctness and numerical robustness that need to be taken into account. The first problem that we encountered appeared when analysing the LPV model presented in (2). It can be observed that a singularity appears around  $v_x = 0$  which is a major problem, as the application requires being able to keep the car still, accelerate and stop it smoothly.

This has been addressed by defining a region where the dynamic model is forced to behave considering only the longitudinal dynamics, which is equivalent to assume a

straight movement. This decision leads to the definition of a warming up/down operating mode bounded by a velocity range between zero and one m/s where lateral dynamics will be neglected in the controller.

Regarding the temporal behaviour of the system there is a hardware limitation in in the CAN Bus communication layer that constraints the PID frequency to 50Hz, restraining the MPC to 10Hz to allow the low level control to reach the steady state. After testing the system under these conditions, it was concluded that the performance of the controller was dramatically reduced due to such a low rate.

It has been decided to implement a control strategy where the controller runs at 30Hz and for each applied control action two extra are simulated. This approach requires to re-sample the planner, as originally it is updated each 100ms. However, due to the lack of extreme driving conditions, it has been decided to “freeze” the reference using a ZOH. The MPC constraints are defined as

$$A_u = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, b_u = \begin{bmatrix} 0.3 \\ 0.3 \\ 15 \\ 5 \end{bmatrix}, \quad (12a)$$

$$A_{\Delta u} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, b_{\Delta u} = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.25 \\ 0.25 \end{bmatrix}, \quad (12b)$$

$$\bar{y}_e = -\underline{y}_e = 3m. \quad (12c)$$

### B. Estimation set up

In this section, we are going to cover the implementation of the estimator in our system. This is motivated by the inherent necessity of obtaining the lateral velocity of the car along with the requirement of cleaning the signals measured, as by nature are very noisy. Firstly, regarding its mathematical formulation we are going to present the set containing the admissible values of the scheduling variables as  $\Theta = \{\phi | A_\phi \phi = b_\phi, \phi \geq 0\}$  being

$$A_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}, b_\phi = \begin{bmatrix} 15 \\ -0.5 \\ 1 \\ 1 \\ 0.3 \\ 0.3 \\ 0.25 \\ 0.25 \end{bmatrix}. \quad (13)$$

It is worth to note that the observer has been designed in continuous time, as it is working at a frequency of 200Hz. This leads to being able to approximate the continuous-time design in a discrete-time framework. It is worth to note that the implementation has been tested in a range between 100 and 200 Hz without noticing any remarkable loss of performance.

Similarly to the controller, singularities appear whenever the system is close to  $v_x = 0$ . In this case, we decided to

use the estimation only when  $v_x = 1m/s$ . As in this range a linear movement is assumed noise in the measurements can be disregarded and the variables that need to be estimated, which are related to lateral displacement, are considered 0.

### C. Framework

The whole system has been implemented under ROS framework to enable proper communication between hardware and different pieces of code. As it was infeasible to use the real car during the testing phase, a simulator was implemented using the dynamic model of the system considering both process and measurement noises. ROS nodes have been programmed using Python and all optimisation is done using OSQP solver [12] [13] [14]. Even taking into account the slower performance of Python compared to other languages like C++ the complete loop is executed in less than 15ms, leaving enough margin for the real system to operate. It is worth to note that trajectories were computed offline.

### D. Results

In this section, we are going to comment the results in a simulated environment of the previously presented system. As our objective is to test its the robustness when operating close to a singularity, the proposed trajectory accelerates from 0 m/s to a cruise velocity of 10 m/s and finally stopping, following an L-shaped track.

All tests have been done simulating trajectories that match the conditions under which the autonomous car is going to be tested. They were done using an MSI GL62M 7RD-429XES (Intel Core i7-7700HQ CPU @ 2.80GHz).

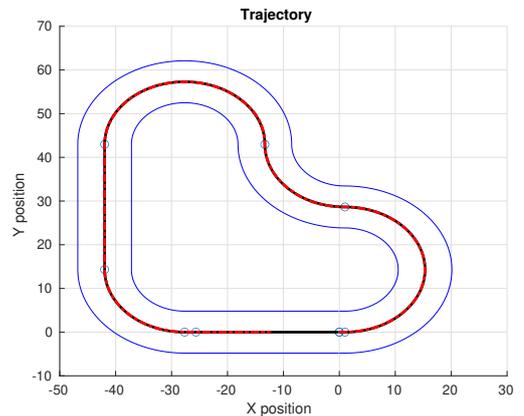


Fig. 3. Trajectory followed during simulation

In this figure, it can be seen how the testing track was defined. Graphically the magnitude of the performance cannot be observed due to the difference of scale between the error and the current circuit. However, it is clear that the controller is able to navigate through the environment without problems. In the following figures, performance is going to be analysed using the errors recorded while performing the experiment. In both states, the errors can be considered small. In the case of the lateral error during the whole experiment, it is below 10cm, which can be considered acceptable as the width of

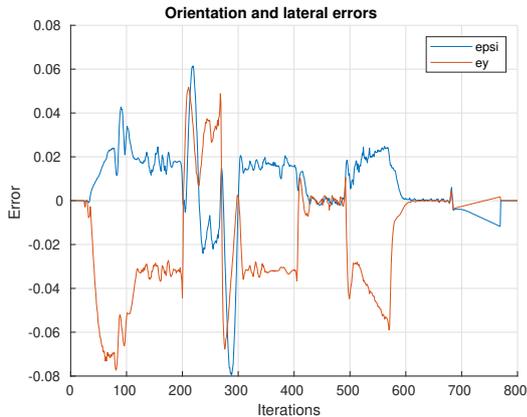


Fig. 4. Orientation and lateral errors

the track 4m. Regarding the error related with the orientation again it appears to be negligible when compared with the magnitude of the heading of the vehicle. When studying the velocity profile we can see that it is followed properly. It is worth to note that the oscillations present in the velocity are expected, as they are derived from having to change the orientation of the vehicle. Finally we are going to evaluate

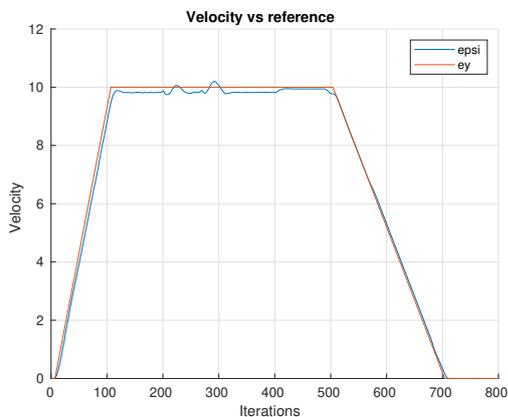


Fig. 5. Following a velocity profile during the experiment

the root mean squared error, which supports the analysis presented in this section.

TABLE II  
RMS ERROR.

	Lateral Error	Heading Error	Velocity Error
RMS	0.0176	0.0070	0.1940

## VI. CONCLUSIONS

An LPV-MPC control scheme has been proposed and implemented to address the autonomous driving problem. To do so, both control and estimation approaches have been designed with a dynamic bicycle model and implemented in a real system, solving the problems related singularities that

are present in this family of models. Finally, the results of this approach have been evaluated in simulation, guaranteeing the feasibility in the real system. The following steps of this research are related with the implementation of this control loop in a physical device, which is already being coded for both a 1:1 Kia Niro and a smaller RC vehicle.

## REFERENCES

- [1] Hu, C., Qin, Y., Cao, H., Song, X., Jiang, K., Rath, J. J., and Wei, C. (2019). Lane keeping of autonomous vehicles based on differential steering with adaptive multivariable super-twisting control. *Mechanical Systems and Signal Processing*, 125, 330-346.
- [2] Xu, S., and Peng, H. (2019). Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*.
- [3] Wang, H., Liu, B., Ping, X., and An, Q. (2019). Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access*, 7, 161064-161073.
- [4] Alcalá, E., Puig, V., Quevedo, J., and Rosolia, U. (2020). Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Engineering Practice*, 95, 104270.
- [5] Besselmann, T., Lofberg, J., and Morari, M. (2012). Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control*, 57(9), 2322-2332.
- [6] Jungers, M., Oliveira, R. C., and Peres, P. L. (2011). MPC for LPV systems with bounded parameter variations. *International Journal of Control*, 84(1), 24-36.
- [7] Sename, O., Gaspar, P., and Bokor, J. (Eds.). (2013). Robust control and linear parameter varying approaches: application to vehicle dynamics (Vol. 437). Springer.
- [8] Osman, M., Hussein, A., and Al-Kaff, A. (2019, September). Intelligent Vehicles Localization Approaches between Estimation and Information: A Review. In 2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES) (pp. 1-8). IEEE.
- [9] Kwiatkowski, A., Boll, M. T., and Werner, H. (2006, December). Automated generation and assessment of affine LPV models. In *Proceedings of the 45th IEEE Conference on Decision and Control* (pp. 6690-6695). IEEE.
- [10] Alcalá, E., Puig, V., Quevedo, J., and Escobet, T. (2018). Gain-scheduling lpv control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory & Applications*, 12(12), 1683-1693.
- [11] Talamino, J. P., and Sanfeliu, A. (2019). Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving. *Robotics and Autonomous Systems*, 114, 93-105.
- [12] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. (2017, November). OSQP: An operator splitting solver for quadratic programs. ArXiv e-prints.
- [13] G. Banjac, P. Goulart, B. Stellato, and S. Boyd. (2019). Infeasibility detection in the alternating direction method of multipliers for convex optimization. *Journal of Optimization Theory and Applications*, 183(2):490-519.
- [14] G. Banjac, B. Stellato, N. Moehle, P. Goulart, A. Bemporad, and S. Boyd. (2017). Embedded code generation using the OSQP solver. In *IEEE Conference on Decision and Control (CDC)*.