Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-28 (2019) 106–113

# LPV-MPC Control for Autonomous Vehicles ⋆

**Eugenio Alcalá[1,2], Vicenç Puig[1,2], Joseba Quevedo[2]**

[1] *Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain*

[2] *Research Center for Supervision, Safety and Automatic Control, UPC, Rambla Sant Nebridi, s/n, 08022 Terrassa, Spain*

Abstract In this work, a novel approach is presented to solve the trajectory tracking problem for autonomous vehicles. This method is based on the use of a cascade control where the external loop solves the position control using a novel Linear Parameter Varying - Model Predictive Control (LPV-MPC) approach and the internal loop is in charge of the dynamic control of the vehicle using a LPV - Linear Quadratic Regulator technique designed via Linear Matrix Inequalities (LPV-LMI-LQR). Both techniques use an LPV representation of the kinematic and dynamic models of the vehicle. The main contribution of the LPV-MPC technique is its ability to calculate solutions very close to those obtained by the non-linear version but reducing significantly the computational cost and allowing the real-time operation. To demonstrate the potential of the LPV-MPC, we propose a comparison between the non-linear MPC formulation (NL-MPC) and the LPV-MPC approach.



*Keywords:* Linear Parameter Varying Systems and Applications , MPC, Autonomous vehicles, Self driving cars

## 1. INTRODUCTION

In the last recent years, we have experienced a great advance in the technological career towards autonomous driving. Today, we can see how research centers and companies in the automotive sectors are accelerating and investing large amounts of money. In addition, if we add to this progress the advances in legislation and the increasing acceptance of the user, we converge on the fact that driving, as we know it today, has days counted. The numerous advantages that the autonomous vehicle offers with respect to traditional vehicles are obvious. However, the most attractive is the great reduction of accidents on the roads, which will lead to a huge reduction in deaths on roads worldwide.

In order to achieve complete autonomous driving, a series of modules are needed working in a sequential and organized manner. First, the vehicle sensing network (GPS, IMU, encoders, cameras, LIDAR, etc) collects all the vehicle and environment information and is treated to extract measurements of interest (vehicle and obstacles position, velocities, etc). Then, the trajectory planning module is responsible for generating the route using the actual vehicle position and the desired one. This trajectory is composed of global positions, orientations and vehicle velocities. Finally, the automatic control generates the control actions (acceleration, steering and braking) for the actuators using the computed sequence of references and the position of the vehicle.

The automatic control is the last piece in the sequence of the autonomous vehicle and one of the most important tasks since it is in charge of guaranteeing its motion. It is also the topic addressed in this paper. From a model-based control point of view, the control problem may be mainly defined by two characteristics: the type of control (lateral, longitudinal or integrated) and the type of model considered for its design (kinematic, linear dynamic, simplified non-linear dynamics or non-linear dynamics). Jiang and Astolfi (2018) and Yang et al. (2017) address the problem of lateral control using non-linear feedback control techniques. Optimal-based techniques like LQR for lateral control problem is formulated in Boyali et al. (2018). Regarding the longitudinal control, we can find LQR strategy in Naeem and Mahmood (2016); Junaid et al. (2005) and $H_2$ in Naeem and Mahmood (2016). However, these control strategies solve simplified versions of the real problem, i.e. the integrated control. This work addresses both the longitudinal-lateral integrated control problem for autonomous vehicles.

⋆ This work has been partially funded by the Spanish Government and FEDER through the projects CICYT DEOCS and SCAV (refs. MINECO DPI2016-76493, DPI2017-88403-R). This work has also been partially funded by AGAUR of Generalitat de Catalunya through the Advanced Control Systems (SAC) group grant (2017 SGR 482), and by AGAUR and the Spanish Research Agency through the Maria de Maetzu Seal of Excellence to IRI (MDM-2016-0656). The author is supported by a FI AGAUR grant (ref 2017 FI B00433).


Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.356

Control strategies based on Linear Parameter Varying (LPV) models aim to solve NL control problems using a pseudo-linear reformulation which incorporates the original non-linearities within new parameters. These parameters depend on some system states and inputs which are called scheduling variables. Some recent books, Tanaka and Wang (2004), Gáspár et al. (2016), Rotondo (2017), Ostertag (2011) and Duan and Yu (2013), presented the study of the modeling and design of LPV under the formulation based on LMI. Several design approaches can be used such as pole positioning, $H_\infty$, $H_2$ and $H_2$-LQR. These techniques have proven to be widely accepted in the field of robotics, for example Rotondo (2017) and Blažič (2017).

Model Predictive Control (MPC) is another technique that has proven to be one of the most interesting methods in this field in recent years. This strategy allows to find the optimal control action through the resolution of a constrained optimization problem in which a mathematical model of the real system is evaluated in a future horizon. Recent articles such as Rawlings and Risbeck (2017), Corriou (2018) and Mayne (2014) present the latest advances in MPC control outside the automotive field. In the field of autonomous vehicles, we can find all kinds of formulations for the MPC. From NL-MPC applications in Ercan et al. (2017), where the lateral control problem is solved, to MPC lateral control using a linearized model of the vehicle in Xu et al. (2017). Working with non-linear models usually gives the best results. However, when working with systems with fast dynamics this technique may result non-viable since its excessive computational time. This is the reason why recent exploration of other ways opens the door to ideas such as Linear Parameter Varying - Model Predictive Control (LPV-MPC). Cisneros et al. (2016) and Besselmann and Morari (2009) present the MPC strategy using LPV models. The advanatge of LPV 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 (Sename et al., 2013).

The contribution of this paper focuses on the use of LPV models for the automatic control strategy design for an autonomous vehicle considering both longitudinal and lateral dynamics. An MPC approach is proposed based on the LPV kinematic formulation of the vehicle that leads to a quadratic optimal problem. In addition, introducing the terminal set concept, we are able to guarantee stability.

The paper is structured as follows: Section 2 gives an overview of the work and describes the different types of modelling used for control purposes. In Section 3, the kinematic and dynamic control designs are developed. Section 4 shows the simulation results and Section 5 presents the conclusions of the work.

## 2. OVERVIEW OF THE PROPOSED SOLUTION

In this work, we consider the problem of urban autonomous guidance. To solve it, two important tasks have to be carried out: the trajectory planning and the automatic control.

On the one hand, the trajectory planning to be followed by the vehicle has to fulfill particular specifications such as continuous and differentiable velocity profiles. Thus, this module is in charge of providing discrete and smooth references to the automatic control stage. On the other hand, the automatic control is in charge of following the planned references, thus, moving the vehicle between two ground coordinates as well as generating smooth control actions for achieving a comfortable journey. In Fig. 1, we show the planning-control diagram proposed in this work. Observe that two control levels have been designed as a cascade scheme, one for the position control and a faster and inner one to control the dynamic behaviour of the vehicle, i.e. linear and angular velocities.

The level of difficulty of a vehicle guidance control problem comes often determined by two aspects: the type of control (lateral, longitudinal or mixed) and the complexity of the model to be controlled (kinematic, linear dynamic, non-linear simplified dynamic or non-linear dynamic). In this work we address one of the most complex configurations, to solve the mixed non-linear dynamic problem. The following subsection covers the formulation of the different models used for solving the control problem.

### 2.1 LPV Control Oriented Models

Unlike controlling common mobile robots which operate at a low interval of velocities, urban cars work in a higher velocities and accelerations range. This fact makes indispensable to study control techniques based on elaborated dynamic models, articularly with the aim of being safer and smoother in the control performance. In this work, two model-based techniques cover the kinematic and dynamic control at different layers and, hence, at different sampling times. For that reason, the use of mathematical kinematic and dynamic vehicle models are necessary. The kinematic model is based on the mass-point assumption while for the dynamic one the bicycle model has been considered. We refer to the Appendix A for the complete model equations used in this paper. In the following subsections, we present the LPV formulation of both, the kinematic and dynamic models.

*Kinematic LPV model.* The state, control and reference vectors, respectively, are denoted as

$$x_c = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}, \; u_c = \begin{bmatrix} v_x \\ \omega \end{bmatrix}, \; r_c = \begin{bmatrix} v_d \cos\theta_e \\ \omega_d \end{bmatrix}, \tag{1a}$$

where $x_e$, $y_e$ and $\theta_e$ are the position and orientation errors, respectively. The inputs $v_x$ and $\omega$ are the longitudinal and angular velocities, respectively. $v_d$ and $\omega_d$ are the longitudinal and angular reference velocities, respectively. Then, defining the vector of scheduling variables as $\rho(k) := [\omega(k), v_d(k), \theta_e(k)]$ which are bounded in $\omega \in [-1.42, 1.42]$ $\frac{rad}{s}$, $v_d \in [0.1, 20]$ $\frac{m}{s}$ and $\theta_e \in [-0.05, 0.05]$ $rad$, the non-linear kinematic model (see Alcala et al. (2018b)) is transformed into the Linear Parameter Varying representation as follows

$$x_c(k+1) = A_c(\rho(k))x_c(k) + B_c u_c(k) - B_c r_c(k), \tag{1b}$$

where

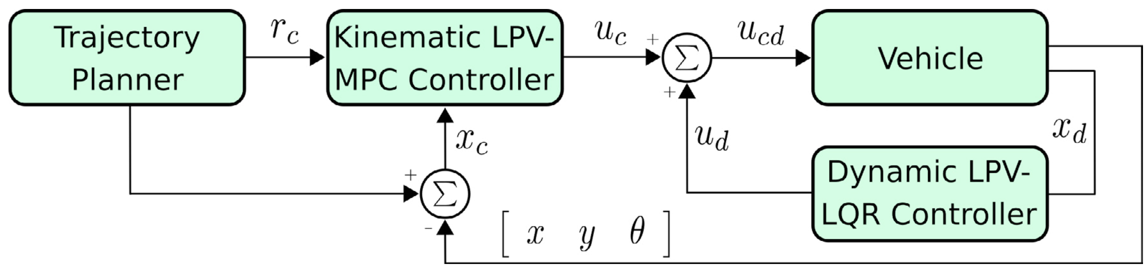


Figure 1. Autonomous guidance scheme composed by the trajectory planning stage and both control layers: kinematic and dynamic.

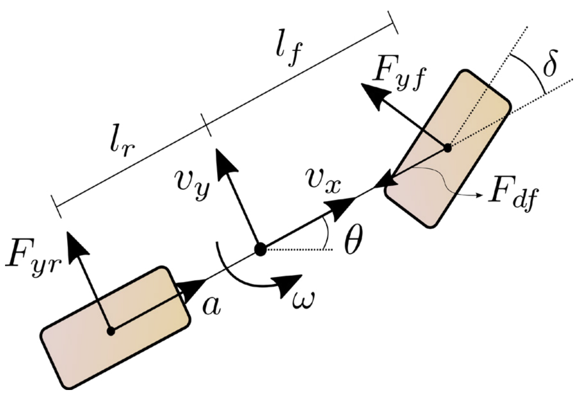


Figure 2. Bicycle model used for control purposes.

$$A_c\left(\rho(k)\right) = \begin{bmatrix} 1 & \omega T_c & 0 \\ -\omega T_c & 1 & v_d \frac{\sin\theta_e}{\theta_e} T_c \\ 0 & 0 & 1 \end{bmatrix} \tag{1c}$$

$$B_c = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} T_c \ , \tag{1d}$$

with $T_c$ being the outer loop sampling time.

From this formulation, a polytopic representation for the control design is obtained as

$$x_c(k+1) = \sum_{i=1}^{2^{n_c}} \mu_i(\rho(k)) A_{c_i} x_c(k) + B_c u_c(k) - B_c r_c(k) \ , \tag{2}$$

being $n_c$ the number of scheduling variables and $A_i$ each one of the polytopic vertex systems obtained as a combination of the extreme values of the scheduling variables.

The expression $\mu_i(\rho(k))$ is known as the membership function and is given by

$$\mu_i(\rho(k)) = \prod_{j=1}^{n_c} \xi_{ij}(\eta_0^j, \eta_1^j) \ , \qquad i = \{1, ..., 2^{n_c}\} \tag{3a}$$

$$\begin{aligned} \eta_0^j &= \frac{\overline{\rho_j} - \rho_j(k)}{\overline{\rho_j} - \underline{\rho_j}} \\ \eta_1^j &= 1 - \eta_0^j \ , \qquad\qquad j = \{1, ..., n_c\} \ , \end{aligned} \tag{3b}$$

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

*Dynamic LPV model.* The dynamic LPV model considered in this work is a transformation of the non-linear one presented in Alcala et al. (2018a).

Then, the state and control vectors are denoted as

$$x_d = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}, \ u_d = \begin{bmatrix} \delta \\ a \end{bmatrix}, \tag{4a}$$

where $v_y$, $a$ and $\delta$ are the lateral velocity, rear wheel longitudinal acceleration and steering angle, respectively (see Fig. 2). The LPV model can be expressed as

$$x_d(k+1) = A_d(\vartheta(k)) x_d(k) + B_d u_d(k) \ , \tag{4b}$$

with the time-varying scheduling vector as $\vartheta(k) := [\delta(k), v_x(k), v_y(k)]$ being $\delta \in [-0.25, 0.25]$ $rad$, $v_x \in [0.1, 20]$ $\frac{m}{s}$ and $v_y \in [-1, 1]$ $\frac{m}{s}$ and

$$A_d(\vartheta(k)) = \begin{bmatrix} 1 + A_{11}T_d & A_{12}T_d & A_{13}T_d \\ 0 & 1 + A_{22}T_d & A_{23}T_d \\ 0 & A_{32}T_d & 1 + A_{33}T_d \end{bmatrix} \tag{4c}$$

$$A_{11} = -\frac{\frac{1}{2} C_d \rho A_r v_x^2 + \mu m g}{m v_x} \ , \ A_{12} = \frac{C_f \sin\delta}{m v_x} \tag{4d}$$

$$A_{13} = \frac{C_f l_f \sin\delta}{m v_x} + v_y \ , \ A_{22} = -\frac{C_r + C_f \cos\delta}{m v_x} \tag{4e}$$

$$A_{23} = -\frac{C_f l_f \cos\delta - C_r l_r}{m v_x} - v_x \tag{4f}$$

$$A_{32} = -\frac{C_f l_f \cos\delta - C_r l_r}{I v_x} \ , \ A_{33} = -\frac{C_f l_f^2 \cos\delta + C_r l_r^2}{I v_x} \tag{4g}$$

$$B_d = \begin{bmatrix} 0 & 1 \\ B_{21} & 0 \\ B_{31} & 0 \end{bmatrix} T_d \tag{4h}$$

$$B_{21} = \frac{C_f}{m} \ , \ B_{31} = \frac{C_f l_f}{I} \ , \tag{4i}$$

with $T_d$ being the sample time used in the dynamic control loop, i.e. the inner control loop. $m$ and $I$ represent the vehicle mass and inertia, respectively. $l_f$ and $l_r$ are the distances from the center of gravity to the front and rear wheel axes, respectively. Variables $C_f$ and $C_r$ represent the tire stifness coefficient for the front and rear wheels. $\rho$, $A_r$, $\mu$ and $C_d$ are the density of the air, the front sectional area of the vehicle, the friction coefficient and the drag coefficient, respectively.

As in the case of kinematic model, we look for a polytopic dynamic formulation like the following

$$x_d(k+1) = \sum_{i=1}^{2^{n_d}} \mu_i\big(\vartheta(k)\big) A_{d_i} x_d(k) + B_{d_i} u_d(k) \ , \qquad (5)$$

being $n_d$ the number of dynamic scheduling variables and $A_{d_i}$ represent each one of the polytopic vertex dynamic systems obtained as a combination of the extreme values of the dynamic scheduling variables. The membership function $\mu_i(\vartheta(k))$ is the same than the one presented in (3) but using the dynamic scheduling vector $\vartheta(k)$.

## 3. CONTROL DESIGN

In this section, we present the control scheme proposed for this work as well as its design. The control strategy of the vehicle has been divided into two nested layers, see Fig 1. The outermost layer controls the vehicle's kinematics, i.e. position and orientation of the car, and works at a frequency of 10 Hz. On the other hand, the internal loop controls the dynamic behavior of the vehicle, i.e. its speeds, at a frequency of 200 Hz. Next, both control loops are described separately.

### *3.1 Kinematic LPV-MPC Design*

At this point, we present the formulation of the LPV-MPC strategy, which focuses on solving position and orientation control of the vehicle.

This strategy is based on the resolution of a linear quadratic optimization problem by using the non-linear kinematic error model in its LPV polytopic representation (2). However, there exist the problem associated with the lack of knowledge of the matrix of scheduling variables through the entire prediction horizon. In Cisneros et al. (2016), the use of the optimized state sequence which is obtained after each optimization is proposed.

In this work, the scheduling variables are states of the system whose desired values are known since the trajectory planner generates them. That is why we propose the use of such references as known scheduling variables for the entire optimization horizon being then the scheduling sequence $\Gamma := [\rho(k), ..., \rho(k+N)]$. In this way, we can compute the evolution of the model more accurately and in anticipation.

In addition, since the basic MPC formulation cannot guarantee the overall stability of the system, we propose the addition of a terminal constraint and a terminal cost to the optimization problem.

To formulate the problem, the polytopic LPV system presented in (2) has been considered. In order to avoid a difficult reading, the sub-index $c$ is omitted in the rest of the subsection. Then, the focus is on a MPC scheme where the cost function is defined as

$$\mathrm{J_k} = \sum_{i=0}^{N-1} \left( x_{k+i}^T Q x_{k+i} + \Delta u_{k+i} R \Delta u_{k+i} \right) + x_{k+N}^T P x_{k+N} \ , \qquad (6)$$

where $Q = Q^T \geq 0$, $R = R^T > 0$ and $P = P^T > 0$ represent the states, input and terminal set tuning matrices of apropriate dimensions, respectively.

At each time $k$ the values of $x_k$ and $u_{k-1}$ are known and the following optimization problem can be solved

$$\begin{aligned}
&\underset{\Delta U_k}{\text{minimize}} \quad \mathrm{J_k}(\Delta U_k, X_k) \\
&\text{subject to} \\
&\qquad x_{k+i+1} = \sum_{j=1}^{2^{r_c}} \mu_j(\rho_{k+i}) A_j x_{k+i} \\
&\qquad\qquad + B u_{k+i} - B r_{k+i} \\
&\qquad u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \ , \quad \forall i = 0, ..., N-1 \\
&\qquad \Delta U_k \in \Delta\Pi \\
&\qquad U_k \in \Pi \\
&\qquad x_{k+N} \in \chi \ ,
\end{aligned} \qquad (7)$$

where

$$\Delta U_k = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N-1} \end{bmatrix} \in \mathbb{R}^m \ , \ U_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix} \in \mathbb{R}^m \ , \qquad (8)$$

being $m$ the number of inputs of the kinematic system. $\Pi$ and $\Delta\Pi$ are the constraint sets for the inputs and their derivatives, respectively.

The set $\chi$ represents the terminal state set. Then, by introducing this constraint in the optimization problem, we force the states to converge into a stable region and then, to ensure the MPC stability. The computation of this terminal safety set is carried out by solving two LMI-based problems.

First, the controller for each polytopic system ($A_i$) is found by solving the following LQR-LMI

$$\begin{bmatrix} Y & (A_i Y + B W_i)^T & Y & W_i^T \\ A_i Y + B W_i & Y & 0 & 0 \\ Y & 0 & Q_{TS}^{-1} & 0 \\ W_i & 0 & 0 & R_{TS}^{-1} \end{bmatrix} \geq 0 \qquad (9)$$
$$\forall i = 1, ..., 2^{r_c} \ ,$$

with $Y = Y^T > 0$, $Q_{TS} = Q_{TS}^T \geq 0$ and $R_{TS} = R_{TS}^T > 0$. This problem returns the matrices $Y$ and $W_i$. Then, the resulting controllers are obtained by $K_i = W_i Y^{-1}$. Note that the terminal set matrix $P$ in (6) is found to be equal to $Y^{-1}$.

This LQR design is a particular formulation for the one presented in Theorem 25 of Tanaka and Wang (2004). The constant nature of kinematic input matrix $B_c$ in (1) allows the use of this simplified LMI version.

The second problem consists on finding the largest terminal region $\chi$. To do so, we solve the following constrained optimization problem using the previously obtained controllers $K_i$

$$\begin{aligned}&\underset{Z}{\text{maximize}} && \mathrm{J_k}(Z)\\&\text{subject to} && \\&&&\begin{bmatrix}-Z & Z(A_i+BK_i)^T\\(A_i+BK_i)Z & -Z\end{bmatrix}<0\\&&&K_iZK_i^T-\overline{u}^2<0 \quad \forall i=1,...,2^{r_c}.\end{aligned} \tag{10}$$

The resulting variable is $Z$. Hence, we compute the largest terminal region as $\chi = \{x|x^T Sx \le 1\}$, with $S = Z^{-1}$. Note that this problem is totally constrained by the maximum values of the control actions.

### 3.2 Dynamic LPV-LQR Design

To design the dynamic controller we use the polytopic system (5). Then, we solve offline the optimal LMI problem (9) for computing the polytope vertex controllers $K_i$. Finally, the dynamic controller gain is computed online following

$$K(\vartheta(k)) = \sum_{i=1}^{2^{r_d}} \mu_i(\vartheta(k))K_i \ , \tag{11}$$

where $\mu_i(\vartheta(k))$ represents the weighting function presented in (3) by using the dynamic scheduling vector defined in (4). The offline computation of polytopic controllers allows this control strategy to work at the desired frequency of 200 Hz.

## 4. SIMULATION RESULT

Table 1. Kinematic LPV-MPC design parameters.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $Q$ | 0.9*diag(0.33 0.33 0.33) | $\overline{u}$ | [1.4 20] |
| $R$ | 0.1*diag(0.8 0.2) | $\underline{u}$ | [-1.4 0.1] |
| $T_c$ | 0.1 $s$ | $\overline{\Delta u}$ | [0.3 2] |
| $N$ | 20 | $\underline{\Delta u}$ | [-0.3 -2] |
| $R_{TS}$ | diag(1 3) | $Q_{TS}$ | diag(1 1 3) |

In this section, we validate the performance of the proposed control scheme through simulation in MATLAB. The considered vehicle for running simulations is an electric RWD one whose dynamics are presented in Appendix B. To show the promising results of the LPV-MPC, we perform a comparison against the non-linear MPC approach (NL-MPC in resulting figures). The LPV-MPC uses planning data to instanciate the state space matrices at every control time step within the MPC prediction.

Then, the optimal control problem (7) is solved at a frequency of 10 Hz using the solver GUROBI (Gurobi (2014)) through YALMIP framework (Lofberg (2004)). This solves the position control problem in a outer loop. Note that the non-linear MPC problem has been computed using IPOPT solver and considering the same adjustment and prediction horizon (see Table 1). In the inner loop, the dynamic state feedback control problem (Section 3.2) is solved at a rate of 200 Hz to follow the velocities provided by the kinematic control.

To verify the real-time feasibility of the presented strategies, we perform the simulations on a DELL Inspiron 15 (Intell core i7-8550U CPU @ 1.80GHzx8). The LPV-MPC, dynamic LPV-LQR and vehicle model parameters are listed in Tables 1, 2 and 4, respectively.

Table 2. Dynamic LPV-LQR design parameters.

| Parameter | Value |
|---|---|
| $Q$ | 0.9*diag(0.66 0.01 0.33) |
| $R$ | 0.1*diag(0.5 0.5) |
| $T_d$ | 0.005 $s$ |

Solving the problem (10) to determine the largest terminal set, we obtain matrix $S$ as

$$S = \begin{bmatrix} 0.465 & 0 & 0 \\ 0 & 23.813 & 76.596 \\ 0 & 76.596 & 257.251 \end{bmatrix}. \tag{12}$$

The tests have been carried out in the circuit of Fig. 3 where the aim is to simulate a road driving at a variable speed. For assessment purposes, a perturbation is introduced in the coefficient of friction, varying it sharply from 1 to 0.5 at $t = 110$ s and from 0.5 to 1 at $t = 120$ s. This scenario intends to show a real case in which the vehicle passes through dry and wet asphalt surface while turning a curve.

Fig. 4 shows both, the linear and angular speed profiles provided by the trajectory planning and the respective vehicle responses for both compared approaches. In Fig. 5, we illustrate the complete set of errors, i.e. $x_e$, $y_e$, $\theta_e$, $v_e$ and $\omega_e$. In both, Fig. 4 and 5, it is seen the close behaviour between LPV-based and NL-based approaches. However, the non-linear MPC is able to better handle the external disturbances.

The respective control actions applied to the simulation vehicle are shown in Fig 6. Note that the LPV-MPC response is as good as the NL-MPC one until the arrival of the disturbance in the longitudinal axis. The steering behaviour is practically the same throughout the test.

An important aspect of control strategies based on optimization is the computational time spent at each optimization procedure. In Fig. 7, we show the elapsed time at each kinematic MPC optimization for both compared methods. Note the computational time improvement achieved when using the LPV-MPC strategy. Finally, a quantitative comparison is made using the root mean squared error (RMSE) as performance measurement as shown in Table 3.

Table 3. Comparison using the root mean squared error measure (RMSE).

| Approach | $x$ | $y$ | $\theta$ | $v$ | $\omega$ |
|---|---|---|---|---|---|
| *LPV-MPC* | 0.589 | 0.238 | 0.016 | 0.302 | 0.014 |
| *NL-MPC* | 0.528 | 0.225 | 0.015 | 0.268 | 0.012 |

These results conclude a similar performance of the LPV-MPC approach in comparison to the NL-MPC version.

## 5. CONCLUSION

In this work, a cascade control scheme (kinematic and dynamic) was presented to solve the problem of integrated control (lateral and longitudinal) of autonomous vehicles.

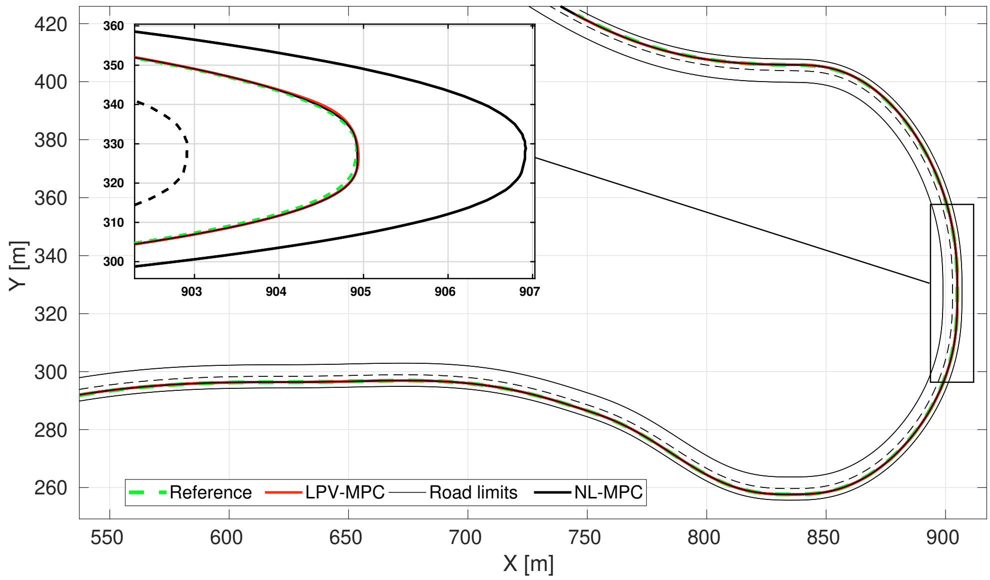


Figure 3. Simulation circuit used for testing the proposed control technique (LPV-MPC) and its non-linear version (NL-MPC).

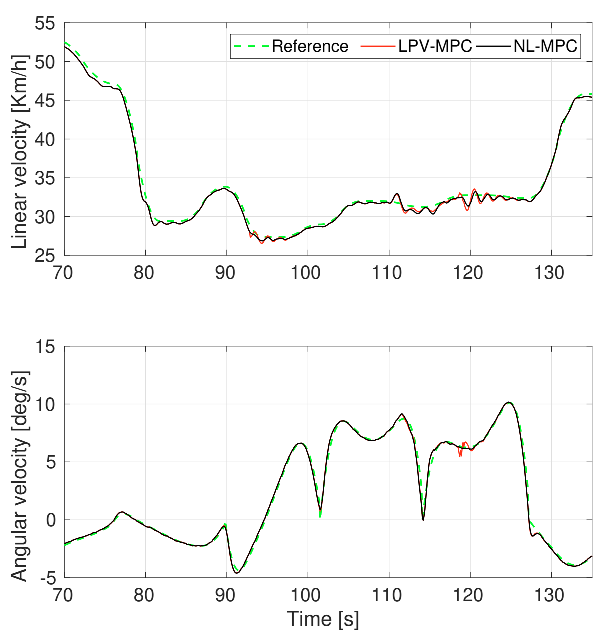


Figure 4. Reference and response velocities for both compared approaches.

The novel kinematic control was designed using the MPC technique with the prediction model expresed in the LPV formulation (LPV-MPC) without using any linearisation. On the other hand, the dynamic control was approached using the Linear Quadratic Regulator (LQR) strategy, with a LPV modeling and using a LMI formulation of the problem (LPV-LMI-LQR).

A comparison was made between two methods of solving the control problem: using the non-linear MPC formulation (NL-MPC) and using updated LPV-MPC with the planner references. It was demonstrated that the LPV-MPC technique works very well compared to the non-linear control problem but in a much faster way (more than 50 times faster).

Table 4. Dynamic model parameters.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $l_f$ | 0.758 $m$ | $A_r$ | 1.91 $m^2$ |
| $l_r$ | 1.036 $m$ | $\rho$ | 1.184 $\frac{kg}{m^3}$ |
| $m$ | 683 $kg$ | $C_d$ | 0.36 |
| $I$ | 560.94 $kg\ m^2$ | $\mu$ | 1 |
| $C_f$ | 24000 $\frac{N}{rad}$ | $C_r$ | 21000 $\frac{N}{rad}$ |
| $d$ | 2680 | $c$ | 1.6 |
| $b$ | 6.1 | | |

## A. VEHICLE MODEL FOR CONTROL

The non-linear equations employed for control purposes are presented as

$$\begin{aligned}
\dot{x}_e &= \omega y_e + v_d \cos\theta_e - v_x \\
\dot{y}_e &= -\omega x_e + v_d \sin\theta_e \\
\dot{\theta}_e &= \omega_d - \omega \\
\dot{v}_x &= a - \frac{F_{yF}\sin\delta}{m} - \frac{F_{df}}{m} + \omega v_y \\
\dot{v}_y &= \frac{F_{yF}\cos\delta}{m} + \frac{F_{yR}}{m} - \omega v_x \\
\dot{\omega} &= \frac{F_{yF} l_f \cos\delta - F_{yR} l_r}{I} \\
F_{yF} &= C_f\left(\delta - \frac{v_y}{v_x} - \frac{l_f\omega}{v_x}\right) \\
F_{yR} &= C_r\left(-\frac{v_y}{v_x} + \frac{l_r\omega}{v_x}\right) \\
F_{df} &= F_{drag} + F_{friction} = \frac{1}{2}C_d\rho A_r v_x^2 + \mu m g
\end{aligned} \tag{13}$$

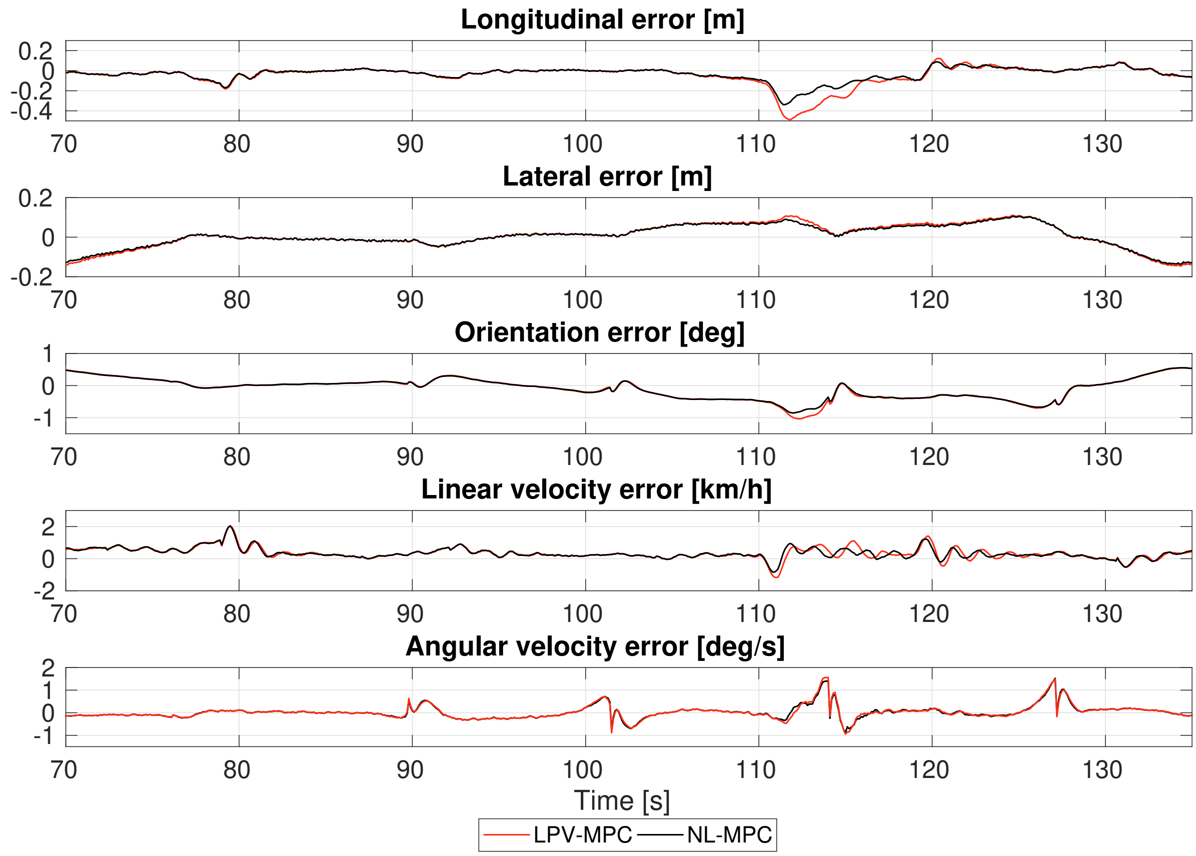


Figure 5. Time evolution of the tracking errors for each compared kinematic control strategy.

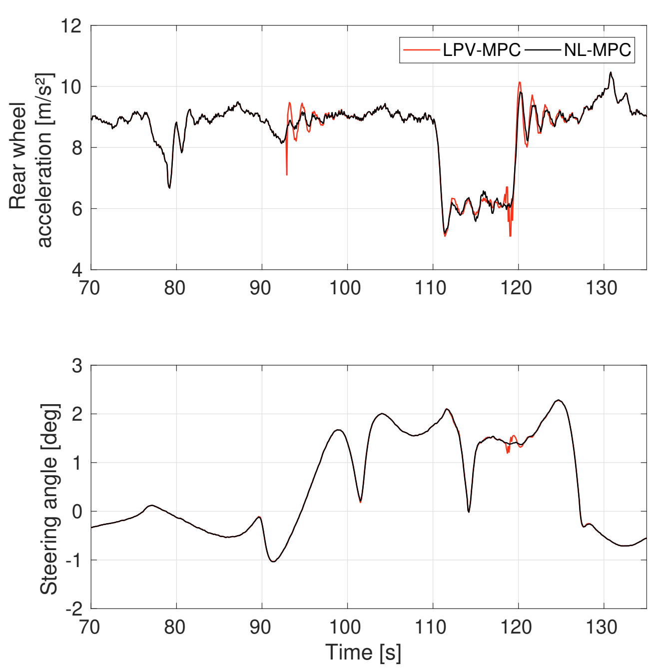


Figure 6. Resulting control actions for both compared methods. Top: traction wheel acceleration. Bottom: steering angle of front wheels.

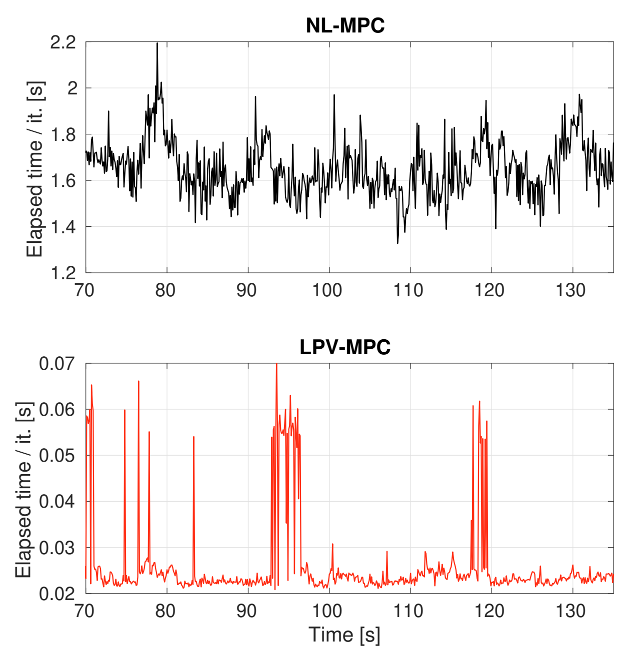


Figure 7. Computational effort comparison when solving the kinematic MPC. Top: Elapsed time per iteration of non-linear MPC strategy. Bottom: Elapsed time per iteration of the proposed LPV-MPC approach.

Refer to the Appendix A in Alcala et al. (2018b) for the complete development of the kinematic error model. Vehicle parameters are defined in Table 4.

## B. VEHICLE MODEL FOR SIMULATION

For simulation purposes we use a higher fidelity vehicle model. Unlike the model used for control design, this considers a more precise tire model, i.e. the Pacejka "Magic Formula" tire model where the parameters $b$, $c$ and $d$ define the shape of the semi-empirical curve. Also, a more accurate computation of the tire slip angles is given.

$$
\begin{aligned}
\dot{x} &= v_x \cos\theta - v_y \sin\theta \\
\dot{y} &= v_x \sin\theta + v_y \cos\theta \\
\dot{\theta} &= \omega \\
\dot{v}_x &= a - \frac{F_{yF}\sin\delta}{m} - \frac{F_{df}}{m} + \omega v_y \\
\dot{v}_y &= \frac{F_{yF}\cos\delta}{m} + \frac{F_{yR}}{m} - \omega v_x \\
\dot{\omega} &= \frac{F_{yF} l_f \cos\delta - F_{yR} l_r}{I} \\
F_{yF} &= d \sin\left(c \tan^{-1}(b\alpha_f)\right) \\
F_{yR} &= d \sin\left(c \tan^{-1}(b\alpha_r)\right) \\
\alpha_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x} - \frac{l_f \omega}{v_x}\right) \\
\alpha_r &= -\tan^{-1}\left(\frac{v_y}{v_x} + \frac{l_r \omega}{v_x}\right) \\
F_{df} &= F_{drag} + F_{friction} = \frac{1}{2} C_d \rho A_r v_x^2 + \mu m g
\end{aligned}
\tag{14}
$$

All parameters are properly defined in Table 4.

## REFERENCES


Alcala, E., Puig, V., Quevedo, J., and Escobet, T. (2018a). Gain scheduling lpv control for autonomous vehicles including friction force estimation and compensation mechanism. *IET Control Theory & Applications*.

Alcala, E., Puig, V., Quevedo, J., Escobet, T., and Comasolivas, R. (2018b). Autonomous Vehicle Control Using a Kinematic Lyapunov-based Technique with LQR-LMI Tuning. *Control Engineering Practice*, 73, 1–12.

Besselmann, T. and Morari, M. (2009). Autonomous vehicle steering using explicit lpv-mpc. In *Control Conference (ECC), 2009 European*, 2628–2633. IEEE.

Blažič, S. (2017). Two approaches for nonlinear control of wheeled mobile robots. In *Control & Automation (ICCA), 2017 13th IEEE International Conference on*, 946–951. IEEE.

Boyali, A., Mita, S., and John, V. (2018). A tutorial on autonomous vehicle steering controller design, simulation and implementation. *arXiv preprint arXiv:1803.03758*.

Cisneros, P.S., Voss, S., and Werner, H. (2016). Efficient nonlinear model predictive control via quasi-lpv representation. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, 3216–3221. IEEE.

Corriou, J.P. (2018). Model predictive control. In *Process Control*, 631–677. Springer.

Duan, G.R. and Yu, H.H. (2013). *LMIs in Control Systems: Analysis, Design and Applications*. CRC press.

Ercan, Z., Gokasan, M., and Borrelli, F. (2017). An adaptive and predictive controller design for lateral control of an autonomous vehicle. In *Vehicular Electronics and Safety (ICVES), 2017 IEEE International Conference on*, 13–18. IEEE.

Gáspár, P., Szabó, Z., Bokor, J., and Németh, B. (2016). Robust control design for active driver assistance systems. *Springer, DOI*, 10, 978–3.

Gurobi, O. (2014). Inc.,"gurobi optimizer reference manual," 2015. *Google Scholar*.

Jiang, J. and Astolfi, A. (2018). Lateral control of an autonomous vehicle. *IEEE Transactions on Intelligent Vehicles*.

Junaid, K.M., Shuning, W., Usman, K., and Naveed, R. (2005). Lqr autonomous longitudinal cruise control with a minimum order state observer. In *Proc. Eighth IASTED Int. Conf., Cambridge, USA*, volume 31.

Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, 284–289. IEEE.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

Naeem, H.M.Y. and Mahmood, A. (2016). Autonomous cruise control of car using lqr and h2 control algorithm. In *Intelligent Systems Engineering (ICISE), 2016 International Conference on*, 123–128. IEEE.

Ostertag, E. (2011). *Mono-and Multivariable Control and Estimation: Linear, Quadratic and LMI Methods*. Springer Science & Business Media.

Rawlings, J.B. and Risbeck, M.J. (2017). Model predictive control with discrete actuators: Theory and application. *Automatica*, 78, 258–265.

Rotondo, D. (2017). *Advances in Gain-Scheduling and Fault Tolerant Control Techniques*. Springer.

Sename, O., Gaspar, P., and Bokor, J. (2013). *Robust control and linear parameter varying approaches: application to vehicle dynamics*, volume 437. Springer.

Tanaka, K. and Wang, H.O. (2004). *Fuzzy control systems design and analysis: a linear matrix inequality approach*. John Wiley & Sons.

Xu, Y., Chen, B., Shan, X., Jia, W., Lu, Z., and Xu, G. (2017). Model predictive control for lane keeping system in autonomous vehicle. In *Power Electronics Systems and Applications-Smart Mobility, Power Transfer & Security (PESA), 2017 7th International Conference on*, 1–5. IEEE.

Yang, J., Bao, H., Ma, N., and Xuan, Z. (2017). An algorithm of curved path tracking with prediction model for autonomous vehicle. In *Computational Intelligence and Security (CIS), 2017 13th International Conference on*, 405–408. IEEE.